(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2004 年3 月4 日 (04.03.2004)

PCT

(10) 国際公開番号
WO 2004/019553 A1

(51) 国際特許分類[7]: H04L 9/32

(21) 国際出願番号: PCT/JP2003/007794

(22) 国際出願日: 2003 年6 月19 日 (19.06.2003)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2002-178947 2002 年6 月19 日 (19.06.2002) JP
特願2003-69375 2003 年3 月14 日 (14.03.2003) JP

(71) 出願人 (米国を除く全ての指定国について): 株式会社エイシーエス (ADVANCED COMPUTER SYSTEMS, INC.) [JP/JP]; 〒211-0005 神奈川県 川崎市 中原区新丸子町915-15 Kanagawa (JP). 株式会社産学連携機構九州 (KYUSHU TLO COMPANY, LIMITED) [JP/JP]; 〒812-8581 福岡県 福岡市 東区箱崎6丁目10番1号 Fukuoka (JP).

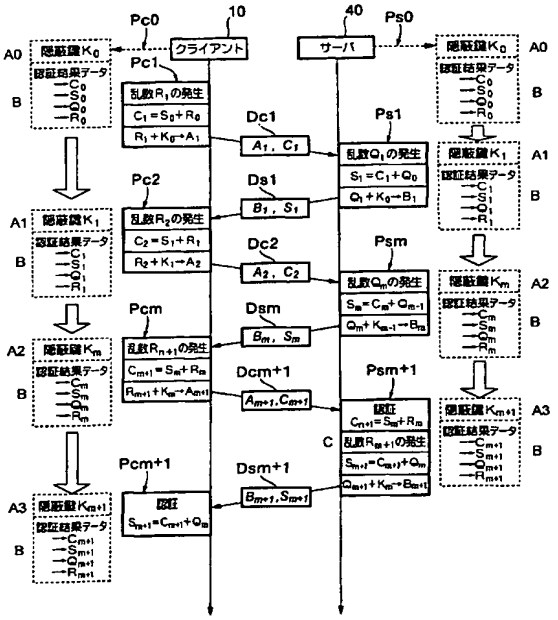
(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 今本 健二 (IMAMOTO,Kenji) [JP/JP]; 〒812-0061 福岡県 福岡市 東区筥松4-22-14 松原寮B-313 Fukuoka (JP). 大河

[続葉有]

(54) Title: INTER-AUTHENTICATION METHOD AND DEVICE

(54) 発明の名称: 相互認証方法及び装置



A0...PRIVATE KEY K_0
B...AUTHENTICATION RESULT DATA
A1...PRIVATE KEY K_1
A2...PRIVATE KEY K_m
A3...PRIVATE KEY K_{m+1}
10...CLIENT
Pc 1...GENERATION OF RANDOM NUMBER R_1
Pc 2...GENERATION OF RANDOM NUMBER R_2
Pcm ...GENERATION OF RANDOM NUMBER R_{n+1}
Pcm +1...AUTHENTICATION
40...SERVER
Ps1...GENERATION OF RANDOM NUMBER Q_1
Psm...GENERATION OF RANDOM NUMBER Q_m
Psm+1...AUTHENTICATION
C...GENERATION OF RANDOM NUMBER Q_{m+1}

(57) Abstract: An inter-authentication method capable of safely and easily performing inter-authentication. In the inter-authentication process, a private key K_0 of the initial value is stored in a client and a server (Pc0, Ps0). The client generates a random number R, calculates password data C and authentication data A, and transmits the result to the server (Pc1). The server receives the authentication data A and the password data C from the client, generates a random number R, calculates and returns password data S and authentication data Q, and updates the private key K_0 to a new private key K_1 (Ps1). The client receives the authentication data B and the password data S from the server, generates a random number R, calculates the password data C_2 and the authentication data A_2, returns the results to the server, and updates the private key K_0 to the new private key K_1 (Pc2). The client and the server check whether validity is satisfied.

[続葉有]

WO 2004/019553 A1

克好 (OKAWA,Katsuyoshi) [JP/JP]; 〒195-0055 東京都 町田市 三輪緑山1-3-2 緑山ヒルズ212 Tokyo (JP). 橋本 努 (HASHIMOTO,Tsutomu) [JP/JP]; 〒184-0002 東京都 小金井市 梶野町5-2-27-206 Tokyo (JP).

(74) 代理人: 酒井 一, 外(SAKAI,Hajime et al.); 〒102-0083 東京都 千代田区 麹町５丁目７番地 秀和紀尾井町ＴＢＲビル Tokyo (JP).

(81) 指定国 (国内): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) 指定国 (広域): ARIPO 特許 (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア特許 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(57) 要約:

安全かつ簡便に相互認証することができる相互認証方法を得る。そのために、相互認証プロセスにおいて、クライアント及びサーバに初期値の隠蔽鍵K_0を格納する（Ｐｃ０，Ｐｓ０）。クライアントは乱数Ｒを生成し暗証データＣ及び認証データＡを計算し、サーバに送信する（Ｐｃ１）。サーバはクライアントから認証データＡ及び暗証データＣを受信しかつ乱数Ｒを生成し暗証データＳ、認証データＱを計算し返信し、隠蔽鍵K_0を新規の隠蔽鍵K_1に更新する（Ｐｓ１）。クライアントはサーバから認証データＢ及び暗証データＳを受信しかつ乱数Ｒを生成し暗証データC_2、認証データA_2を計算しサーバに返信し、隠蔽鍵K_0を新規の隠蔽鍵K_1に更新する（Ｐｃ２）。クライアント及びサーバは正当性が成立するか検査する。

明　細　書

相互認証方法及び装置

技術分野

本発明は、ネットワークに接続されたコンピュータシステム等の装置における相互認証方法及び装置並びにこれに用いるワンタイムＩＤの生成方法、認証方法、認証システム、サーバ、クライアントおよびプログラム関するものである。

とりわけ、本発明は、少なくとも第１認証装置と第２認証装置との間の関係についての正当性を検証する相互認証方法及び装置、並びに複数の装置間またはアプリケーション間における認証に用いて好適なワンタイムＩＤの生成方法、上記ワンタイムＩＤを用いた認証方法、認証システム、サーバ、クライアントおよびプログラムに関するものである。

ネットワーク上において、あるユーザが自分の身元を証明するためには、認証が必要である。認証とは、何らかのプロトコルにより、証明者が自分の身元を検証者に対して証明することであり、電子商取引などの分野において、必須の技術である。例えば、ユーザがサーバに対して、身元を証明したいときは、ユーザが証明者に、サーバが検証者に対応する。また、逆に、サーバがユーザに対して身元を証明したい場合は、サーバが証明者に、ユーザが検証者に対応する。１対１の装置の間では、その立場が反転する場合があるので、相互に認証する相互認証が必要である。

相互認証は、ユーザとサーバの間に限定されず、任意のコンピュータ間での身元を証明する方法として、幅広く利用されている。最近では、公開鍵暗号を用いたものが知られており、証明者は公開鍵と秘密鍵を所持しており、証明者が公開鍵に対応する秘密鍵を所持することを、なんらかのプロトコルにより、検証者に示すことにより、身元を証明している。

しかしながら、従来の相互認証方法では、認証に用いる鍵が単一であるため、ひとたび鍵が知られてしまうと、第三者がユーザになりすまして認証される場合がある。また、ユーザは、鍵の保管に注意を払わなければならず、簡便に利用す

ることができなかった。

例えば、インターネットのような非同期型ネットワークでは、複数のコンピュータが同時に通信しており、証明者が複数の検証者と同時にプロトコルを実行する場合がある。WWW（Ｗｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ：ワールド・ワイド・ウェブ）では、ＨＴＴＰ（Ｈｙｐｅｒｔｅｘｔ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ：WWWサーバとWWWブラウザやＷｅｂブラウザ等が、ファイル等の情報授受に使うプロトコル）のサーバと、接続先であるクライアントとの間では、多数の認証が要求される。

また、上記のような相互認証技術において、従来より、ネットワークを介してコンピュータ間（例えば、クライアント・サーバ間）で通信を行う際には、不正なアクセス等を排除するために、サービス等の提供に先立って認証が行われる。この認証においては、第三者が知り得ない所定の秘密情報（例えば、ＩＤやパスワード、乱数、或いはそれら情報を引数とする関数値など）を予め双方が共有し、その秘密情報に基づいて各々の正当性を相互に検証するのが一般的である。

他方、ＩＥＴＦ（Internet Engineering Task Force）が公式に発行するＲＦＣ（Request For Comments）においては、インターネットでＩＰパケットの暗号化と認証を行なうセキュリティプロトコルとして、ＩＰｓｅｃ（Security Architecture for Internet Protocol）が規定されている。このＩＰｓｅｃでは、暗号・認証のパラメータを動的に生成して交換するＩＫＥ(Internet Key Exchange）という自動鍵交換のプロトコルが標準的に採用されている（例えば、特開２００２－３７４２３８号公報（段落番号０００２～０００９）参照）。

そして、近年では、このＩＫＥの方式にワンタイムＩＤを導入して、既知共有鍵を用いたＩＫＥの方式で問題となっていた、ＩＤ情報保護、ＤｏＳ（Denial of Service attack）攻撃防止、リモートアクセスなどを実現したＰ－ＳＩＧＭＡと呼ばれる鍵交換・認証方式が提案されている。

このＰ－ＳＩＧＭＡにおいては、例えば、図１に示すような手順で鍵交換および認証が行われている。

先ず、クライアントが、ＳＡ（Security Association）の提案、乱数Ｒｃ、Ｄ

H（Diffie-Hellman）公開値g^x、ＯＩＤ（ワンタイムＩＤ）をサーバに対して送信する。なお、ＳＡの提案には、暗号アルゴリズムや認証方式、鍵交換に使用するパラメータ等に関する提案が含まれている。

次いで、サーバが、受け取ったＯＩＤからクライアントを識別し、識別できない場合には、通信を拒否する。識別できる場合には、受諾したＳＡ、乱数Ｒｓ、ＤＨ公開値g^y、ＨＡＳＨｓ、セッション鍵ｅで暗号化したＩＤｓ（サーバＩＤ）をクライアントに対して送信する。なお、セッション鍵ｅは、既知共有鍵、乱数Ｒｓ、乱数ＲｃおよびＤＨ共通鍵g^{xy}を引数とする鍵付きハッシュ関数の関数値であり、ＨＡＳＨｓは、既知共有鍵、乱数Ｒｓ、乱数Ｒｃ、ＤＨ公開値g^x、g^yおよびＩＤｓを引数とする疑似乱数関数の関数値である。

次いで、クライアントが、受け取ったＨＡＳＨｓを検証し、このＨＡＳＨｓに基づいてサーバの正当性を確認する。ＨＡＳＨｓが正しければ、ＨＡＳＨｃ、セッション鍵ｅで暗号化したＩＤｃ（クライアントＩＤ）をサーバに対して送信する。ここで、ＨＡＳＨｃは、既知共有鍵、乱数Ｒｓ、乱数Ｒｃ、ＤＨ公開値g^x、g^yおよびＩＤｃを引数とする疑似乱数関数の関数値である。

次いで、サーバが、受け取ったＨＡＳＨｃを検証し、このＨＡＳＨｃに基づいてクライアントの正当性を確認する。ＨＡＳＨｃが正しければ、当該プロトコルを終了する。

このＰ－ＳＩＧＭＡにおいて、ＯＩＤ（ワンタイムＩＤ）は、次のように定義されている。

ＯＩＤ1 ＝ｐｒｆ（Ｋ、１）
ＯＩＤ2 ＝ｐｒｆ（Ｋ、２）
・・・
ＯＩＤｎ ＝ｐｒｆ（Ｋ、ｎ）

・・・・・（式１）

この定義式において、ＯＩＤｎ はｎ番目のＳＡ確立時に用いられるワンタイムＩＤ、ｐｒｆは疑似乱数関数、Ｋは既知共有鍵、若しくは既知共有鍵から生成された値である。

このため、上記Ｐ－ＳＩＧＭＡによれば、ＯＩＤを導入したことにより、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればＯＩＤを識別情報として把握できるといった効果が得られるとともに、クライアント・サーバ間で通信が行われる度（すなわち、ＳＡの生成または更新毎）にＯＩＤが変更されるため、第三者が次回のＯＩＤを予測することができないといった効果が得られる。

しかしながら、上記Ｐ－ＳＩＧＭＡにおいては、既知共有鍵がひとたび知られてしまうと、すべてのＯＩＤが予測されてしまい、その結果、ＯＩＤの将来にわたる安全性（すなわち、ＰＦＳ：Perfect Forward Security）を保証できなくなるといった問題点があった。

以上、具体例として、Ｐ－ＳＩＧＭＡと呼ばれる鍵交換・認証方式について述べてきたが、一般に、ワンタイムＩＤを用いて複数の装置間またはアプリケーション間における認証を行う認証方式では、特定の秘密情報に基づいてすべてのワンタイムＩＤの生成を行っており、上記同様の問題点を有している。

本発明は、上記従来技術における種々の不具合事実を考慮してなされたもので、その第１の目的は、安全かつ簡便に相互認証することができる相互認証方法及び装置を得ることである。

本発明の第２の目的は、盗聴が困難で安全性に優れたワンタイムＩＤの生成方法、上記ワンタイムＩＤを用いた認証方法、認証システム、サーバ、クライアントおよびプログラムを提供することである。

発明の開示

上記目的を達成するために本発明は、通信回線を介して接続された第１認証装置と第２認証装置の相互関係を認証する相互認証方法であって、前記第１認証装置を特定するための記憶データと、第２認証装置を特定するための記憶データとを、前記第１認証装置及び第２認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを用いて更新した更新結果を履歴データとして、前記第１認証装置及び第２認証装置の各々に共通に記憶する記憶工程を含む

ものである。前記第1認証装置は、記憶されている履歴データを用いて記憶データを新規に生成しかつ生成した新規の記憶データを前記履歴データを用いて暗号化して第2認証装置に送信する第1送信工程と、前記第2認証装置からの記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第1更新工程とを含み、また、前記第2認証装置は、前記第1認証装置からの記憶データ及び記憶されている履歴データを用いて新規に記憶データを生成しかつ生成した新規の記憶データを前記履歴データを用いて暗号化して第1認証装置に送信する第2送信工程と、前記第1認証装置からの記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第2更新工程とを含む。そして、前記第1認証装置及び第2認証装置の少なくとも一方の装置において、履歴データに基づいて記憶データの正当性が成立したときに、第1認証装置と第2認証装置の相互関係が正当であると検証する。

本発明はまた、上記相互認証方法を実施するための相互認証装置を実現するものである。この相互認証装置は、通信回線を介して接続された第1認証装置と第2認証装置とから成り、前記第1認証装置と第2認証装置との間の相互関係を認証するものであって、前記第1認証装置に設けられ当該第1認証装置を特定するための記憶データを格納する第1のメモリと、前記第2認証装置に設けられ当該第2認証装置を特定するための記憶データを格納する第2のメモリと、前記第1認証装置及び第2認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを格納する認証データ記憶手段と、前記認証データを用いて更新した更新結果を履歴データとして、前記第1認証装置及び第2認証装置の各々に共通に記憶する履歴データ記憶手段と、前記第1認証装置又は第2認証装置のうち、認証用データ送信側の認証装置に設けられ、前記履歴データを用いて記憶データを新規に生成する記憶データ生成手段と、生成した新規の記憶データを前記履歴データを用いて暗号化して認証用データ受信側の認証装置に送信する第1の送信手段と、認証用データ受信側の認証装置に設けられ、前記認証用データ送信側の認証装置からの記憶データ及び記憶されている履歴データを用いて新規に記憶データを生成する記憶データ生成手段と、生成した新規の記憶データを

前記履歴データを用いて暗号化して前記認証用データ送信側の認証装置に返信する第２の送信手段と、認証用データ送信側の認証装置に設けられ、前記認証用データ受信側の認証装置から返信された記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第１の更新手段と、認証用データ受信側の認証装置に設けられ、前記認証用データ送信側の認証装置からの記憶データ及び前記返信した新規の記憶データによって前記履歴データを更新する第２の更新手段とを含み、前記第１認証装置及び第２認証装置の少なくとも一方の装置において、前記履歴データに基づいて記憶データの正当性が成立したときに、第１認証装置と第２認証装置の相互関係が正当であると検証する検証手段とを備えている。

かかる相互認証装置は、生成した新規の記憶データを前記履歴データを用いて暗号化するための認証用データを演算する演算手段を有することができる。また、この相互認証装置は、前記演算手段により認証用データを生成するに際し、暗号化用のデータを生成する乱数発生手段を有することも可能である。

また、本発明では、第１認証装置を特定するための記憶データと、第２認証装置を特定するための記憶データとを、履歴データとして、前記第１認証装置及び第２認証装置の各々に共通に記憶する。この履歴データは、第１認証装置及び第２認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを用いて更新した更新結果である。第１認証装置は、記憶されている履歴データを用いて新規の記憶データを生成しかつ新規の記憶データを記憶されている履歴データを用いて暗号化して第２認証装置に送信する。これを第２認証装置が受け取り、第２認証装置は第１認証装置からの記憶データ及び記憶されている履歴データを用いて新規の記憶データを生成しかつ新規の記憶データを記憶されている履歴データを用いて暗号化して第１認証装置に送信する。このとき、第１認証装置は、第２認証装置からの記憶データ及び送信した新規の記憶データによって履歴データを更新する。また、第２認証装置は、第１認証装置からの記憶データ及び送信した新規の記憶データによって履歴データを更新する。この送信工程の後は、第１認証装置及び第２認証装置の少なくとも一方の装置において、履

歴データに基づいて記憶データの正当性が成立したときに、第１認証装置と第２認証装置の相互関係が正当であると検証する。すなわち、第１認証装置及び第２認証装置の一方の認証装置では、他方の認証装置からの履歴を含むデータを受け取り、記憶されている履歴データと照合することが可能となる。そして送信するときには、記憶されている履歴データから新規で異なる履歴データに基づくデータを送信するので、同一データによる授受はない。このため、秘匿性を向上させることもできる。

より詳細には、前記履歴データを履歴データＫとして、該履歴データＫとして記憶する、前記第１認証装置を特定するための記憶データは、暗証データＣ及び認証データＲであり、前記第２認証装置を特定するための記憶データは、暗証データＳ及び認証データＱであることを特徴とする。

前記第１送信工程は、記憶されている履歴データＫの暗証データＳ及び認証データＲを用いて暗証データＣを新規に生成しかつ、記憶されている履歴データＫの認証データＲについて新規に生成し、生成した新規の認証データＲを前記履歴データＫを用いて暗号化して認証データＡを求め、前記認証データＡ及び新規の暗証データＣを第２認証装置に送信し、前記第１更新工程は、前記第２認証装置からのデータを受信し、前記送信した新規の暗証データＣ、受信した新規に生成された暗証データＳ、受信した新規に生成された認証データＱ、及び前記送信した新規の認証データＲにより、前記履歴データＫを更新し、前記第２送信工程は、前記第１認証装置からのデータを受信し、受信した新規の暗証データＣ及び記憶されている履歴データＫの認証データＱを用いて暗証データＳを新規に生成しかつ記憶されている履歴データＫの認証データＱについて新規に生成し、生成した新規の認証データＱを記憶した履歴データＫを用いて暗号化して認証データＢを求め、前記認証データＢ及び新規の暗証データＳを第１認証装置に送信し、前記第２更新工程は、受信した新規の暗証データＣ、新規に生成した暗証データＳ、新規に生成した認証データＱ、及び受信した新規の認証データＲにより、前記履歴データＫを更新し、前記第１認証装置及び第２認証装置の少なくとも一方の装置において、履歴データＫに基づいて暗証データの正当性が成立したときに、第

1認証装置と第2認証装置の相互関係が正当であると検証することを特徴とする。

前記記憶工程は、前記第1送信工程、第1更新工程、第2送信工程、及び第2更新工程における認証による更新結果を履歴データとして記憶することを特徴とする。

前記認証データR及び認証データQの少なくとも一方は、乱数発生手段により発生された乱数、データ容量、時間データの少なくとも1つであることを特徴とする。

前記第1認証装置の第1送信工程では、前記暗証データS及び認証データRによる予め定めた関数の演算結果の値を暗証データCとして生成し、前記第2認証装置の第2送信工程では、前記暗証データC及び前記認証データQによる予め定めた関数の演算結果の値を暗証データSとして生成することを特徴とする。

前記第1認証装置の第1送信工程では、前記生成した新規の認証データR及び前記履歴データKによる予め定めた関数の演算結果の値を認証データAとして求め、前記第2認証装置の第2送信工程では、前記生成した新規の認証データQ及び前記履歴データKによる予め定めた関数の演算結果の値を認証データBとして求めることを特徴とする。

前記第1認証装置の検証工程は、前記履歴データKのうち記憶されている認証データQ及び前回送信する前に生成した暗証データCによる予め定めた関数の演算結果の値が受信した暗証データSと一致するときに前記相互関係が正当であると検証することを特徴とする。

前記第2認証装置の検証工程は、前記履歴データKのうち記憶されている暗証データS及び認証データRによる予め定めた関数の演算結果の値が受信した暗証データCと一致するときに前記相互関係が正当であると検証することを特徴とする。

前記記憶工程は、前記第1送信工程、第2送信工程、第1更新工程及び第2更新工程を複数実施した結果、得られるデータを履歴データKとして記憶することを特徴とする。

以上の説明から明らかなように、本発明によれば、第1認証装置及び第2認証

装置の間で相互認証するときに、第１認証装置及び第２認証装置の各々に共通に履歴データを記憶すると共に、履歴データを更新するので、安全かつ簡便に相互認証することができ、例えば、クライアント・コンピュータとサーバ・コンピュータとの間で授受される情報から、クライアント・コンピュータの鍵が漏洩することがなく、確実に認証が行えるという効果がある。

本発明はまた、上記相互認証方法及び装置において使用されるワンタイムＩＤの生成方法を提供する。これは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするものである。

ここで、一方向関数とは、引数から結果（関数値）を求めるのは簡単であるが、結果から引数を求めるのは難しい関数のことを云い、この一方向関数には、例えば、ハッシュ関数、疑似乱数関数などが含まれる。

所定の通信単位としては、例えば、ＩＰｓｅｃにおいてＳＡが確立されてから当該ＳＡが無効になるまでの間にクライアント・サーバ間で行われる一連の通信を、所定の通信単位として設定することも可能であるし、装置間またはアプリケーション間で行われる１回のデータ送受信を所定の通信単位として設定することも可能である。

可変共有鍵は、上記所定の通信単位毎に変化し、且つ認証を行う装置間またはアプリケーション間で共有される、第三者が知り得ない秘密情報であれば、如何なる鍵であってもよい。

認証とは、一方の装置（または一方のアプリケーション）が他方の装置（または他方のアプリケーション）にアクセスする際に、他方の装置が一方の装置の正当性を確認することを云い、識別情報とは、上記認証において少なくとも一方の装置から他方の装置に送信されて当該他方の装置が一方の装置を識別するのに用

いる情報（ＩＤ）のことを云う。

また、上記認証には、一方の装置が他方の装置の認証を行う一方向認証と、双方の装置で相互に認証を行う相互認証とが含まれる。例えば、上記認証においてワンタイムＩＤを使用する方法としては、双方の装置でワンタイムＩＤを生成するとともに、一方の装置が他方の装置にワンタイムＩＤを送信し、他方の装置が、一方の装置から受信したワンタイムＩＤと自らが生成したワンタイムＩＤとの比較・照合により、一方の装置を識別或いは認証する方法が挙げられる。

本発明は、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするものである。

本発明はまた、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするものである。

本発明はまた、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムＩＤを送信して、他方の装置が、一方の装置から受信したワンタイムＩＤと自らが生成したワンタイムＩＤとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムＩＤを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、

この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするものである。

本発明はまた、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムＩＤを送信して、他方の装置が、一方の装置から受信したワンタイムＩＤと自らが生成したワンタイムＩＤとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムＩＤを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするものである。

本発明はまた、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムＩＤを送信して、他方の装置が、一方の装置から受信したワンタイムＩＤと自らが生成したワンタイムＩＤとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムＩＤを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤ（ＳＩＧＮＡＬｎ）を用いて、互いに通信を行なう第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、上記第二装置との間で予め共有化された可変共有鍵を用いて上記ワンタイムＩＤを生成するとともに、この生成したワンタイムＩＤと、当該第一装置に予め設定さ

れたＩＤを少なくとも引数とする一方向関数Ｆｃの関数値と、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方とを上記第二装置に対して送信するステップと、上記第二装置が、上記ワンタイムＩＤおよび上記一方向関数Ｆｃの関数値を演算により求め、この演算結果と、上記第一装置から受信したワンタイムＩＤおよび一方向関数Ｆｃの関数値との照合により、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、当該第二装置に予め設定されたＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とを上記第一装置に対して送信するステップと、上記第一装置が、上記一方向関数Ｆｓの関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数Ｆｓの関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とするものである。

本発明はまた、上記認証方法において、上記一方向関数Ｆｃとして、所定の共有鍵、上記Diffie-Hellman公開値の一方、上記第一装置に予め設定されたＩＤ、上記ワンタイムＩＤを引数とする疑似乱数関数を用いるとともに、上記一方向関数Ｆｓとして、上記所定の共有鍵、上記Diffie-Hellman公開値の一方、上記Diffie-Hellman公開値の他方、上記第二装置に予め設定されたＩＤ、上記ワンタイムＩＤを引数とする疑似乱数関数を用いることを特徴とするものである。

本発明はまた、可変共有鍵を生成し、且つ可変共有鍵と通信順序に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いて、第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、上記第二装置との間で予め共有化された第一の可変共有鍵と当該第一装置の通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬn､j）として生成するとともに、上記第一の可変共有鍵を用いて、当該第一装置に予め設定されたＩＤ、上記第二装置に予め設定されたＩＤ、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムＩＤを暗号化し、この暗号化データと上記第一のワンタイムＩＤとを上記第二装置に対して送信するステップと、上

記第二装置が、上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムＩＤとの照合により、上記第一装置を識別するステップと、上記第二装置が、上記第一装置を識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたＩＤ、当該第二装置に予め設定されたＩＤおよび上記第一のワンタイムＩＤに基づいて、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、上記第一の可変共有鍵と当該第二装置の通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬ’ｎ､１）として生成するとともに、上記第一装置から受信したDiffie-Hellman公開値の一方と当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記第一装置に予め設定されたＩＤ、当該第二装置に予め設定されたＩＤおよび上記第二のワンタイムＩＤを引数とする一方向関数ｈの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムＩＤとを上記第一装置に対して送信するステップと、上記第一装置が、上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムＩＤとの照合により、上記第二装置を識別するステップと、上記第一装置が、上記第二装置を識別できた場合に、上記第二装置から受信した上記Diffie-Hellman公開値の他方と当該第一装置に予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数ｈの関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数ｈの関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とするものである。

本発明はまた、上記認証方法において、上記第二のワンタイムＩＤを生成する一方向関数として、上記第一のワンタイムＩＤを生成する一方向関数とは異なる一方向関数を用いるようにしたことを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、

所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いて、第一装置と第二装置間における認証（相互認証）を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬc_1）として求め、この第一のワンタイムＩＤと上記第一の乱数とを上記第二装置に対して送信するステップと、上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬs_1）として求め、この第二のワンタイムＩＤと上記第二の乱数とを上記第一装置に対して送信するステップと、上記第一装置が、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記第二装置から受信した上記第二のワンタイムＩＤとの比較により、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムＩＤ（ＳＩＧＮＡＬc_2）として求め、この第三のワンタイムＩＤを上記第二装置に対して送信するステップと、上記第二装置が、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記第一装置から受信した上記第三のワンタイムＩＤとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いて、第一装置と第二装置間における認証（相互認証）を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置

との間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬc_1）として求め、この第一のワンタイムＩＤと上記第一の乱数を上記第二装置に対して送信するステップと、上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬs_1）として求め、この第二のワンタイムＩＤと上記第二の乱数を上記第一装置に対して送信するステップと、上記第一装置が、上記第一の乱数および上記共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記第二装置から受信した上記第二のワンタイムＩＤとの比較により、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワンタイムＩＤ（ＳＩＧＮＡＬc_2）として求め、この第三のワンタイムＩＤを上記第二装置に対して送信するステップと、上記第二装置が、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記第一装置から受信した上記第三のワンタイムＩＤとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とするものである。

本発明はまた、上記認証方法において、上記第一の乱数と上記第二の乱数を、上記第一装置と上記第二装置との間で予め共有化された共有鍵で暗号化した状態で、送信するようにしたことを特徴とするものである。

本発明はまた、上記認証方法において、上記第二装置が上記第二のワンタイムＩＤと上記第二の乱数とを上記第一装置に対して送信するステップにおいて、上記第二装置は、上記第一装置との間で予め共有化された乱数を初期乱数として、この初期乱数と上記第一の乱数を引数とする所定の演算を行い、この演算結果を上記第一装置に対して送信する一方、上記第一装置は、上記第二装置の正当性の判定材料として、上記第二装置から受信した上記演算結果を、上記第二のワンタイムＩＤとともに用いることを特徴とするものである。

本発明はまた、上記認証方法において、上記第一装置が上記第三のワンタイムＩＤを上記第二装置に対して送信するステップにおいて、上記第一装置は、上記

第一の乱数と上記第二の乱数を引数とする所定の演算を行い、この演算結果を上記第二装置に対して送信する一方、上記第二装置は、上記第一装置の正当性の判定材料として、上記第一装置から受信した上記演算結果を、上記第三のワンタイムＩＤとともに用いることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いて、第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬci）として求め、当該第一装置に予め設定されたＩＤ、上記第二装置に予め設定されたＩＤおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムＩＤとを上記第二装置に対して送信するステップと、上記第二装置が、上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムＩＤとの照合により、上記第一装置を識別するステップと、上記第二装置が、上記第一装置を識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたＩＤおよび当該第二装置に予め設定されたＩＤに基づいて、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬsi）として求め、上記第一装置に予め設定されたＩＤ、当該第二装置に予め設定されたＩＤおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムＩＤとを上記第一装置に対して送信するステップと、上記第二装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップと、上記第一装置が、上記第二のワンタイムＩＤを演算により求め、この演算結果と、上

記第二装置から受信した上記第二のワンタイムＩＤとの照合により、上記第二装置を識別するステップと、上記第一装置が、上記第二装置を識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる、上記第二装置に予め設定されたＩＤおよび当該第一装置に予め設定されたＩＤに基づいて、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップとを有することを特徴とするものである。

本発明はまた、上記認証方法において、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後に、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようにしたことを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤ（ＳＩＧＮＡＬｎ）を用いてクライアントとの間で認証を行うサーバであって、上記クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムＩＤとを上記クライアントから受信する受信手段と、上記一方向関数の関数値Ｆｃおよび上記ワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムＩＤおよび上記一方向関数Ｆｃの関数値との比較により、上記クライアントの正当性を判定する判定手段と、上記判定手段が上記クライアントを正当であると判定した場合に、当該サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、当該サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する送信手段とを備えることを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤ（ＳＩＧＮＡＬｎ）を用いてサーバとの間で認証を行

うクライアントであって、上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムＩＤを生成するとともに、当該クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値を演算により求め、これらワンタイムＩＤおよび一方向関数Ｆｃの関数値と、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する送信手段と、上記サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する受信手段と、上記一方向関数Ｆｓの関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数Ｆｓの関数値との比較により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするものである。

本発明はまた、認証システムとして、上記サーバと、上記クライアントとを備えてなることを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤ（ＳＩＧＮＡＬｎ）に基づいてクライアントとの間で認証を行うサーバに実行させるプログラムであって、上記クライアシトに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムＩＤとを上記クライアントから受信する処理と、上記一方向関数の関数値Ｆｃおよび上記ワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムＩＤおよび上記一方向関数Ｆｃの関数値との比較により、上記クライアントの正当性を判定する処理と、上記クライアントが正当であると判定された場合に、上記サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する処理とを上記サーバに実行させることを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有

鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤ（ＳＩＧＮＡＬｎ）に基づいてサーバとの間で認証を行うクライアントに実行させるプログラムであって、上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムＩＤを生成するとともに、上記クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値を演算により求め、これらワンタイムＩＤおよび一方向関数Ｆｃの関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する処理と、上記サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する処理と、上記一方向関数Ｆｓの関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数Ｆｓの関数値との比較により、上記サーバの正当性を判定する処理とを上記クライアントに実行させることを特徴とするものである。

本発明はまた、可変共有鍵を生成し、且つ可変共有鍵と通信順序に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いてクライアントとの間で認証を行うサーバであって、上記クライアントとの間で予め共有化された第一の可変共有鍵と上記クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬｎ､ｊ）として、この第一のワンタイムＩＤ、上記クライアントに予め設定されたクライアントＩＤ、当該サーバに予め設定されたサーバＩＤ、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方を上記第一の可変共有鍵で暗号化した暗号化データと、上記第一のワンタイムＩＤとを上記クライアントから受信する受信手段と、上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムＩＤとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記クライアントＩＤ、上記サーバＩＤおよび上記第一のワンタイムＩＤに基づいて、上記クライアントの正当性を判定する判定

手段と、上記判定手段が上記クライアントを正当であると判定した場合に、上記第一の可変共有鍵と当該サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬ’ｎ、$_1$）として生成するとともに、上記クライアントから受信したDiffie-Hellman公開値の一方と当該サーバに予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記クライアントＩＤ、上記サーバＩＤおよび上記第二のワンタイムＩＤを引数とする一方向関数ｈの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムＩＤとを上記クライアントに対して送信する送信手段とを備えることを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いてサーバとの間で認証を行うクライアントであって、上記サーバとの間で予め共有化された第一の可変共有鍵と当該クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬｎ、ｊ）として生成するとともに、上記第一の可変共有鍵を用いて、当該クライアントに予め設定されたクライアントＩＤ、上記サーバに予め設定されたサーバＩＤ、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムＩＤを暗号化し、この暗号化データと上記第一のワンタイムＩＤとを上記サーバに対して送信する送信手段と、上記第一の可変共有鍵と上記サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬ’ｎ、$_1$）とし、Diffie-Hellman共通鍵を第二の可変共有鍵として、上記第二のワンタイムＩＤ、上記第二の可変共有鍵、上記クライアントＩＤおよび上記サーバＩＤを引数とする一方向関数ｈの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方と、上記第二のワンタイムＩＤとを上記サーバから受信する受信手段と、上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムＩＤとの照合により、上記サーバを識別し、上記サーバを識別

した場合に、上記サーバから受信した上記Diffie-Hellman公開値の他方と当該クライアントに予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数ｈの関数値を演算により求め、この演算結果と、上記サーバから受信した一方向関数ｈの関数値との照合により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするものである。

本発明はまた、認証システムを、上記サーバと、上記クライアントとから構成したことを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬc_1）として、この第一のワンタイムＩＤと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬs_1）として求め、この第二のワンタイムＩＤと上記第二の乱数とを上記クライアントに対して送信する送信手段と、上記第一の乱数、上記第二の乱数および第二の共有鍵を引数とする一方向関数の関数値を第三のワンタイムＩＤ（ＳＩＧＮＡＬc_2）として、この第三のワンタイムＩＤを上記クライアントから受信する第二受信手段と、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムＩＤとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、

所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いてサーバとの間で相互に認証を行うクライアントであって、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬc_1）として求め、この第一のワンタイムＩＤと上記第一の乱数とを上記サーバに対して送信する第一送信手段と、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬs_1）として、この第二のワンタイムＩＤと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムＩＤとの比較により、上記サーバの正当性を判定する判定手段と、上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムＩＤ（ＳＩＧＮＡＬc_2）として求め、この第三のワンタイムＩＤを上記サーバに対して送信する第二送信手段とを備えることを特徴とするものである。

本発明はまた、認証システムとして上記サーバと、上記クライアントとを備えてなることを特徴とするものである。

本発明はまた、請求項２９に記載の発明は、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬc_1）として、この第一のワンタイムＩＤと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、第二の乱数を生成するととも

に、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬs_1）として求め、この第二のワンタイムＩＤと上記第二の乱数を上記クライアントに対して送信する送信手段と、上記共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムＩＤ（ＳＩＧＮＡＬc_2）として、この第三のワンタイムＩＤを上記クライアントから受信する第二受信手段と、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムＩＤとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いてサーバとの間で相互に認証を行うクライアントであって。第１の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬc_1）として求め、この第一のワンタイムＩＤと上記第一の乱数を上記サーバに対して送信する第一送信手段と、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬs_1）として、この第二のワンタイムＩＤと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、上記第一の乱数および上記共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムＩＤとの比較により、上記サーバの正当性を判定する判定手段と、上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワンタイムＩＤ（ＳＩＧＮＡＬc_2）として求め、この第三のワンタイムＩＤを上記サーバに対して送信する第二送信手段とを備えることを特徴とするものである。

本発明はまた、認証システムとして上記サーバと、上記クライアントとを備え

てなることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬci）として、この第一のワンタイムＩＤを上記クライアントから受信するとともに、上記クライアントで生成された第一の乱数、上記クライアントに予め設定されたクライアントＩＤ、当該サーバに予め設定されたサーバＩＤを上記共有鍵で暗号化した第一の暗号化データを上記クライアントから受信する受信手段と、上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムＩＤとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる上記クライアントＩＤおよび上記サーバＩＤに基づいて、上記クライアントの正当性を判定する判定手段と、上記判定手段が上記クライアントを正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬsi）として求め、上記クライアントＩＤ、上記サーバＩＤおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムＩＤとを上記クライアントに対して送信する送信手段と、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いてサーバとの間で相互に認証を行うクライアントであって、第一

の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムＩＤ（ＳＩＧＮＡＬci）として求め、当該クライアントに予め設定されたクライアントＩＤ、上記サーバに予め設定されたサーバＩＤおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムＩＤとを上記サーバに対して送信する送信手段と、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムＩＤ（ＳＩＧＮＡＬsi）として、この第二のワンタイムＩＤを上記サーバから受信するとともに、上記サーバで生成された第二の乱数、上記クライアントＩＤおよび上記サーバＩＤを上記共有鍵で暗号化した第二の暗号化データを上記サーバから受信する受信手段と、上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムＩＤとの照合により、上記サーバを識別し、上記サーバを識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる上記サーバＩＤおよび上記クライアントＩＤに基づいて、上記サーバの正当性を判定する判定手段と、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするものである。

本発明はまた、認証システムとして上記サーバと、上記クライアントとを備えてなることを特徴とするものである。

本発明はまた、上記認証システムにおいて、上記サーバおよび上記クライアントは、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後で、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようになっていることを特徴とするものである。

本発明に係る、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤ（ＳＩＧＮＡＬn）を用いて、互いに通信を行なう第一装置と第二装置間における認証を行う認証方法によれば、可変共有鍵を引数と

する一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成するようにしたため、例えば、可変共有鍵が第三者に漏れたとしても、所定の通信単位毎に可変共有鍵が変化することとなるので、漏れた可変共有鍵を用いて生成されたワンタイムＩＤ以外のワンタイムＩＤを予測することはできない。すなわち、盗聴が困難で安全性に優れたワンタイムＩＤを生成することが可能になり、ワンタイムＩＤの将来にわたる安全性（ＰＦＳ）を実現することが可能になる。

本発明ではまた、可変共有鍵を生成し、且つ可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いて、第一装置と第二装置間における認証を行うようにしたため、例えば、可変共有鍵が第三者に漏れたとしても、所定の通信単位毎に可変共有鍵が変化するとともに、各通信毎に通信順序または回数に関する情報も変化することとなるので、漏れた可変共有鍵を用いて生成されたワンタイムＩＤ以外のワンタイムＩＤを予測することは事実上不可能となり、また漏れた可変共有鍵を用いて生成されたワンタイムＩＤの予測自体も非常に困難なものとなる。すなわち、盗聴が困難で安全性に優れたワンタイムＩＤを生成することが可能になり、ワンタイムＩＤの将来にわたる安全性（ＰＦＳ）を実現することが可能になる。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、このワンタイムＩＤを用いて、第一装置と第二装置間における認証（相互認証）を行うようにしたため、例えば、共有鍵が第三者に漏れたとしても、乱数によって一方向関数の関数値が所定の通信単位毎に変化することとなるので、所定の通信単位内で生成される乱数がわからない限り、ワンタイムＩＤを予測することはできない。すなわち、盗聴が困難で安全性に優れたワンタイムＩＤを生成することが可能になり、ワンタイムＩＤの将来にわたる安全性（ＰＦＳ）を実現することが可能になる。

本発明はまた、上述の種々のワンタイムＩＤの生成方法により生成されたワン

タイムＩＤを用いて、装置間（クライアント・サーバ間）における認証を行うようにしたので、第三者（攻撃者）が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムＩＤを識別情報として把握することができる。

したがって、ＤｏＳ攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ＩＤ情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることができる。

本発明ではまた、第一装置の正当性を判定するのに用いる一方向関数Ｆｃとして、所定の共有鍵、Diffie-Hellman公開値の一方、第一装置に予め設定されたＩＤ、ワンタイムＩＤを引数とする疑似乱数関数を用いるとともに、第二装置の正当性を判定するのに用いる一方向関数Ｆｓとして、所定の共有鍵、Diffie-Hellman公開値の一方、Diffie-Hellman公開値の他方、第二装置に予め設定されたＩＤ、ワンタイムＩＤを引数とする疑似乱数関数を用いるようにしたので、従来の鍵交換・認証方式では３回必要であった通信回数を２回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

本発明はさらに、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成したり、可変共有鍵を生成し、且つ可変共有鍵と通信順序に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、装置間またはアプリケーション間で所定の可変共有鍵を生成したり、或いは、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成するなどの種々のワンタイムＩＤの生成方法により生成されたワンタイムＩＤを用いて、装置間（クライアント・サーバ間）における認証を行うようにしたので、盗聴が困難で安全性に優れたワンタイムＩＤを生成することが可能となり、ワンタイムＩＤの将来にわたる安全性（ＰＦＳ）を実現できるといった効果が得られる。

本発明はさらに、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共

有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成したり、可変共有鍵を生成し、且つ可変共有鍵と通信順序に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成し、装置間またはアプリケーション間で所定の可変共有鍵を生成したり、或いは、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムＩＤを生成するなどの種々のワンタイムＩＤの生成方法により生成されたワンタイムＩＤを用いて、装置間（クライアント・サーバ間）の認証を行うようにしたので、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムＩＤを識別情報として把握できるといった効果が得られる。

したがって、ＤｏＳ攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ＩＤ情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることもできる。

かかる本発明の目的及び利点は添付図面を参照して説明される、以下の実施の形態によってより一層明らかになるであろう。

図面の簡単な説明

図１は、Ｐ－ＳＩＧＭＡと呼ばれる従来の認証方法を説明する図である。

図２は、本発明の実施の形態に係るクライアント・コンピュータとサーバ・コンピュータとの概略構成を示すブロック図である。

図３は、本発明の実施の形態に係る相互認証における概念プロセスを示すフローチャートである。

図４は、本発明の実施の形態に係る相互認証における詳細プロセスを示すイメージ図である。

図５は、本発明に係る認証システムの一実施形態を示す概略構成図である。

図６は、図１のサーバの概略構成を示すブロック図である。

図７は、図１のクライアントの概略構成を示すブロック図である。

図８は、本発明に係る認証方法の第１の実施形態を説明する図である。

図９は、本発明に係る認証方法の第２の実施形態を説明する図である。

図１０は、本発明に係る認証方法の第３の実施形態を説明する図である。

図１１は、本発明に係る認証方法の第４の実施形態を説明する図である。

図１２は、本発明に係る認証方法の第５の実施形態を説明する図である。

図１３は、本発明に係る認証方法の第６の実施形態を説明する図である。

図１４は、ＯＳＰＡと呼ばれる従来の認証方法を説明する図である。

図１５は、本発明に係る認証方法の第７の実施形態を説明する図である。

図１６は、図１５の変形例を説明する図である。

発明を実施するための最良の形態

（実施の形態１）

以下、図面を参照して本発明の実施の形態の一例を詳細に説明する。図２は本発明の第１の実施の形態に係るクライアント・コンピュータとサーバ・コンピュータとの概略構成、および本発明が適用可能なネットワーク・システムの概略構成を示すブロック図である。本実施の形態は、ネットワークにおいてサーバ・コンピュータとクライアント・コンピュータとの間で相互認証する場合に本発明を適用したものである。

図２において、ネットワークシステムは、ＣＰＵを少なくとも含む一または複数のクライアント・コンピュータ１０、及びＣＰＵを少なくとも含む一または複数のサーバ・コンピュータ４０が、それぞれモデム、ルータ、ＴＡ（ターミナル・アダプタ：Ｔｅｒｍｉｎａｌ　Ａｄａｐｔｅｒ）等を介して、ネットワーク（例えば、インターネット）３２に接続されて構成されている。これらのコンピュータは、ネットワーク３２を介して、相互通信により情報授受が可能である。

なお、図２に示すように、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の各々は１つのコンピュータとして説明するが、これらのクライアント・コンピュータ１０、サーバ・コンピュータ４０は複数台でもよい。

なお、クライアント・コンピュータ１０が本発明の第１認証装置に相当するとき、サーバ・コンピュータ４０が第２認証装置に相当し、サーバ・コンピュータ４０が本発明の第１認証装置に相当するとき、クライアント・コンピュータ１０

が第２認証装置に相当する。また、ネットワーク３２は本発明の通信回線に相当する。

本実施の形態では、ネットワークとしてインターネットを適用した場合を説明する。この場合、少なくとも１つのコンピュータは、WWWサーバとして機能させることができ、また他のマシンはWWWクライアントとして機能させることもできる。

詳細には、各クライアント・コンピュータ１０には、WWWブラウザがインストールされており、このWWWブラウザを起動することにより、ネットワーク３２を介してサーバ・コンピュータ４０に任意にアクセス可能となる。このとき、アクセス位置（アクセス先のサーバ・コンピュータ４０の位置、及びサーバ・コンピュータ４０内の情報の位置で構成されるデータ）は、ＵＲＬ（Ｕｎｉｆｏｒｍ　Ｒｅｓｏｕｒｃｅ　Ｌｏｃａｔｏｒ）で指定される。

サーバ・コンピュータ４０は、クライアント・コンピュータ１０からアクセス要求があった場合、ＵＲＬで指定された位置にあるデータを、ネットワーク３２を介して、アクセス元のクライアント・コンピュータ１０へ送信する。このとき、データは、一般に、ＨＴＴＰに従って転送される。

なお、クライアント・コンピュータ１０の識別には、ＩＰ（Ｉｎｔｅｒｎｅｔ　Ｐｒｏｔｏｃｏｌ）アドレスが用いられる。また、クライアント・コンピュータ１０を操作するユーザの識別には、ユーザ自身の入力や、予め定められているコード等のユーザＩＤを用いることができる。

上記コンピュータには、当該コンピュータで指示入力をするために、各々キーボード、マウス等の入力装置が設けられており、コンピュータによる処理結果等を表示するためにディスプレイが設けられている。なお、コンピュータは、汎用的かつ一般的なハードウェア構成であるため、詳細な説明を省略する。

クライアント・コンピュータ１０は、システムパラメータ等を入力するための入力装置１２を備えており、入力装置１２は入力に応じた乱数Ｒを発生する乱数発生器１４及びメモリ１６に接続されている。乱数発生器１４は、メモリ１６及び乱数Ｒに基づく認証用データＡを求める認証用データ演算器１８に接続されて

いる。認証用データ演算器１８は、ネットワーク３２を介してサーバ・コンピュータ４０と通信するためにネットワーク３２に接続された通信インタフエース（以下、通信Ｉ／Ｆという）３０に接続されている。

通信Ｉ／Ｆ３０には、検証器２０が接続されている。この検証器２０はメモリ１６及び認証用データ演算器１８にも接続されている。また、検証器２０は、サーバ・コンピュータ４０との間で認証したときに、認証により相互関係が正当であると判定されたことを表示するＯＫ装置２２及び認証により相互関係が不当であると判定されたことを表示するＮＧ装置２４にも接続されている。

サーバ・コンピュータ４０は、システムパラメータ等を入力するための入力装置４２を備えており、入力装置４２は入力に応じた乱数Ｑを発生する乱数発生器４４及びメモリ４６に接続されている。乱数発生器４４は、メモリ４６及び乱数Ｒに基づく認証用データＢを求める認証用データ演算器４８に接続されている。認証用データ演算器４８は、ネットワーク３２を介してクライアント・コンピュータ１０と通信するために通信Ｉ／Ｆ６０に接続されている。

通信Ｉ／Ｆ６０には、検証器５０が接続されている。この検証器５０はメモリ４６及び認証用データ演算器４８にも接続されている。また、検証器５０は、クライアント・コンピュータ１０との間で認証したときに、認証により相互関係が正当であると判定されたことを表示するＯＫ装置５２及び認証により相互関係が不当であると判定されたことを表示するＮＧ装置５４にも接続されている。

〔概念プロセス〕

次に、本実施の形態のネットワーク・システムにおける相互認証の概念プロセスを説明する。本実施の形態では、コンピュータ間の相互認証をデジタルデータの授受で実行する。図３には、相互認証の処理プロセスをフローチャートとして示した。

ステップ１００では、クライアント・コンピュータ１０及びサーバ・コンピュータ４０は、予め定めた手順により、双方に共通な初期値（隠蔽鍵Ｋ。）を記憶する。

予め定めた手順とは、クライアント・コンピュータ１０及びサーバ・コンピュ

ータ４０の間の相互認証を実行するときの初期値を設定するものである。例えば、クライアント・コンピュータ１０及びサーバ・コンピュータ４０に共通なデータを初期値として保持させるため、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の何れか一方、または第三者のコンピュータによって定められる初期値を、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方へ提供する。この提供は、初期値を電子メールなどの電子的にデータ送信することや、初期値を印刷した印刷物をクライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方に送付してクライアント・コンピュータ１０及びサーバ・コンピュータ４０の各々で入力することによって達成される。

本実施の形態では、この初期値として、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方で共通な状態を維持するため、クライアント・コンピュータ１０とサーバ・コンピュータ４０との間でなされるデータ授受の履歴を初期値とし、後のクライアント・コンピュータ１０とサーバ・コンピュータ４０との間でなされるデータ授受毎に初期値を更新する。

すなわち、上記初期値は、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方に共通な値であればよく、上記のように任意の値を提供することで双方で保持してもよいが、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方で共通な状態を維持するため、任意のアルゴリズムによるクライアント・コンピュータ１０及びサーバ・コンピュータ４０の間のデータ授受の結果が好ましい。本実施の形態では、任意のアルゴリズムには、送信側と受信側との双方のデータを送信側及び受信側の双方で共通に保持する手順で可能であり、詳細を後述する相互認証の結果のデータを用いている。

なお、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方に記憶するデータの形式（例えばフォーマット）は、同一に限定するものではない。すなわち、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方に記憶するデータは、そのデータの最終的な値が同一であればよく、データそのものの同一性に限定されない。例えば、異なる形式で格納するようにしてもよい。このようにすれば、一方のデータが漏洩した場合であっても、他方のデ

ータは維持可能となる。

まず、ステップ１１０では、クライアント・コンピュータ１０が、認証データを送付する。この認証データは、クライアント・コンピュータ１０からサーバ・コンピュータ４０に対して相互認証を要求する最初のデータであり、記憶されている初期値を隠蔽鍵として用い、クライアント・コンピュータ１０内で生成されるデータを記憶すると共に隠蔽鍵による暗号化を行って、送付する。

次に、ステップ１２０では、サーバ・コンピュータ４０において、クライアント・コンピュータ１０から送付された認証データを受け取って、記憶されている初期値を隠蔽鍵として用い、この時点でサーバ・コンピュータ４０内で生成されるデータを記憶すると共に隠蔽鍵による暗号化を行った認証データを送付する。なお、認証データには、クライアント・コンピュータ１０から受け取った認証データに含まれる一部のデータを含ませる。

これにより、サーバ・コンピュータ４０から送付する認証データがクライアント・コンピュータ１０からの要求に対する応答であることを表すデータとして送付することができる。この認証データを送付した後には、受け取った認証データを解析すると共に、サーバ・コンピュータ４０内で生成したデータの各々を用いて新規の隠蔽鍵を生成すると共に、新規の隠蔽鍵で、記憶されている隠蔽鍵を更新する。

次に、ステップ１３０では、クライアント・コンピュータ１０において、サーバ・コンピュータ４０から送付された認証データを受け取って、記憶されている初期値を隠蔽鍵として用い、この時点でクライアント・コンピュータ１０内で生成されるデータを記憶すると共に隠蔽鍵による暗号化を行った認証データを送付する。なお、認証データには、サーバ・コンピュータ４０から受け取った認証データに含まれる一部のデータを含ませる。

これにより、クライアント・コンピュータ１０から送付する認証データがサーバ・コンピュータ４０から送付されたものに対する応答であることを表すデータとして送付することができる。この認証データを送付した後には、受け取った認証データを解析すると共に、クライアント・コンピュータ１０内で生成したデー

タの各々を用いて新規の隠蔽鍵を生成すると共に、新規の隠蔽鍵で、記憶されている隠蔽鍵を更新する。

従って、ステップ１３０のプロセスが終了した時点で、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方において、初期値（隠蔽鍵）が更新されて、共通の値（隠蔽鍵）として維持することができる。

次のステップ１４０では、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方のプロセスが予め定めた所定回数を完了したか否かを判断する。この判断基準回数は、少なくとも１回の回数が予め設定されており、本実施の形態では、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方に共通の回数の値が保持される。なお、判断基準回数は、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の各々で異なる回数の値を保持してもよい。この場合には、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の各々で認証の基準が異なることになるが、認証が正当であれば判断基準回数が少ないコンピュータ側で複数回のデータ授受が要求されることのみで達成できる。この回数を参照することで、クライアント・コンピュータ１０では、ステップ１４０の更新処理、サーバ・コンピュータ４０ではステップ１２０の更新処理が保持されている回数を終了するまで否定される。判断基準回数が１回に設定されている場合には、ステップ１４０で否定されることなく、そのままステップ１５０へ進む。

従って、ステップ１４０で肯定判断された時点で、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方において、共に値（隠蔽鍵）が更新されて、双方で共通の値（隠蔽鍵）が維持されることになる。すなわち、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方で保持する隠蔽鍵が情報授受毎に新規なものに更新され、常時最新の隠蔽鍵として維持することができる。

ステップ１５０では、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方において、認証処理が実行されて本プロセスを終了する。

上記認証処理は、記憶されている最新の隠蔽鍵を用いて、送付された認証デー

タが正当かデータであるか否かを判別することによってなされる。この認証処理は、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方において共通に実行できる。この認証処理が完了すると、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方において相互認証が完了したことになる。

〔詳細プロセス〕

次に、上記概念プロセスで述べた相互認証を詳細に説明する。

（隠蔽鍵を含むデータの構成）

本実施の形態では、隠蔽鍵は、情報授受毎に最新データに更新されるため、履歴データＫとして機能する。以下の説明では、この履歴データＫとして機能するものとして隠蔽鍵Ｋを同一表記とする。

上記概念プロセスで認証データとして用いる初期値を含む隠蔽鍵Ｋは、クライアント・コンピュータ１０を特定するための暗証データＣ及び認証データＲと、サーバ・コンピュータ４０を特定するための暗証データＳ及び認証データＱと、から構成される。なお、以下の説明では、隠蔽鍵Ｋ、暗証データＣ、認証データＲ、暗証データＳ及び認証データＱに初期値「０」から増加する添え字を付し、更新状態を表すものとするが、これらを一般的に説明する場合には添え字を削除した記号のみを用いて説明する。

本実施の形態では、初期値として、詳細を後述するクライアント・コンピュータ１０及びサーバ・コンピュータ４０の双方でなされたデータ授受の結果を記憶するものとし、既に履歴データが内在するものとする。

隠蔽鍵Ｋは、暗証データＣ、認証データＲ、暗証データＳ及び認証データＱの各々を用いた関数ｇ（Ｃ，Ｓ，Ｑ，Ｒ）の計算結果を用いる。関数ｇは、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。

また、クライアント・コンピュータ１０側の初期値Ｃ。，Ｒ。を生成するための最初の値は、暗証データＣ及び認証データＲについてユーザが設定した値を用いてもよく、自動的に生成してもよい。認証データＲは情報授受毎に内容が無規則で変動することが好ましいので、本実施の形態では、認証データＲとして乱数

発生器１４で発生された乱数を用いている。しかし、本発明は、認証データＲに乱数を用いることに限定されるものではない。例えば、現在年月日、日時、時刻などの時間データ、コンピュータ内に格納された任意ファイル容量やタイムスタンプ、及び情報授受のときの容量などを用いることができる。

同様に、サーバ・コンピュータ４０側の初期値Ｓ。，Ｑ。を生成するための最初の値は、暗証データＳ及び認証データＱについてサーバ・コンピュータ４０を管理するオペレータが設定した値を用いてもよく、自動的に生成してもよい。上記と同様に認証データＱは情報授受毎に内容が無規則で変動することが好ましいので、本実施の形態では、認証データＱとして乱数発生器４４で発生された乱数を用いている。しかし、本発明は、認証データＱに乱数を用いることに限定されるものではない。例えば、現在年月日、日時、時刻などの時間データ、コンピュータ内に格納された任意ファイル容量やタイムスタンプ、及び情報授受のときの容量などを用いることができる。

また、クライアント・コンピュータ１０側の認証データＲ、及びサーバ・コンピュータ４０側の認証データＱを他方へ送信するが、その送信データについて第三者による特定を困難にするため、秘匿する必要がある。そこで、本実施の形態では、クライアント・コンピュータ１０からサーバ・コンピュータ４０へ送信する認証データＲ、及びサーバ・コンピュータ４０からクライアント・コンピュータ１０へ送信する認証データＱを隠蔽鍵Ｋで隠蔽する。

すなわち、クライアント・コンピュータ１０からサーバ・コンピュータ４０へ送信する場合、予め定めた関数ｖ（Ｒ，Ｋ）により認証データＡを生成して送信する。関数ｖは、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。同様に、サーバ・コンピュータ４０からクライアント・コンピュータ１０へ送信する場合も、予め定めた関数ｗ（Ｑ，Ｋ）により認証データＢを生成して送信する。関数ｗは、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。次に、関数ｖ、ｗの一例を示す。

$A_m = v(R, K) = R_m + K_{m-1}$

$B_m = w(Q, K) = Q_m + K_{m-1}$

ただし、ｍ≧１の自然数である。

また、クライアント・コンピュータ１０側の暗証データＣ、及びサーバ・コンピュータ４０側の暗証データＳを他方へ送信するが、以下に説明するように、暗証データは情報授受の度に変更している。すなわち、クライアント・コンピュータ１０からサーバ・コンピュータ４０へ送信する暗証データＣは、その送信するときに予め定めた関数ｙ（Ｓ，Ｒ）により新規の暗証データＣを生成して送信する。関数ｙは、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。同様に、サーバ・コンピュータ４０からクライアント・コンピュータ１０へ送信する場合も、予め定めた関数ｚ（Ｃ，Ｑ）により暗証データＳを生成して送信する。関数ｚは、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。次に、関数ｙ、ｚの一例を示す。

$C_m = y(S, R) = S_{m-1} + R_{m-1}$

$B_m = w(C, Q) = C_{m-1} + Q_{m-1}$

ただし、ｍ≧１の自然数である。

なお、暗証データの送信では、第三者による特定を困掛こするため、秘匿してもよい。例えば、クライアント・コンピュータ１０からサーバ・コンピュータ４０へ送信する暗証データＣ、及びサーバ・コンピュータ４０からクライアント・コンピュータ１０へ送信する暗証データＳを隠蔽鍵Ｋで隠蔽するようにしてもよい。すなわち、隠蔽鍵Ｋをパラメータとして追加した関数にしてもよい。

（詳細プロセス）

図４は本発明の第１の実施の形態に係る相互認証における詳細プロセスを示すイメージ図である。以下、図４を参照して本実施の形態の詳細プロセスを説明する。

ステップＰ０：クライアント・コンピュータ１０及びサーバ・コンピュータ４０の各々に、初期値の隠蔽鍵K_0を格納する。このプロセスは、図３のステップ１００と、図４のプロセスＰｃ０及びＰｓ０に相当する。

ステップＰ１：クライアント・コンピュータ１０では、乱数Ｒを生成し、暗証データＣ及び認証データＡを計算し、サーバ・コンピュータ４０に送信する。こ

のプロセスは、図３のステップ１１０と、図４のプロセスＰｃ１に相当する。

すなわち、クライアント・コンピュータ１０においては、乱数発生器１４において乱数R_1を生成する。生成された乱数R_1、メモリ１６に記憶されている隠蔽鍵K_0及び隠蔽鍵K_0を構成するC_0，S_0，Q_0，R_0は認証用データ演算器１８へ入力される。そして、認証用データ演算器１８は、その乱数R_1及びメモリ１６に記憶されている隠蔽鍵K_0及び隠蔽鍵K_0を構成する暗証データS_0、認証データR_0を用いて上記関数ｙ、ｖにより新規の暗証データC_1、及び新規の認証データA_1を求める。この求めた新規の暗証データC_1、及び認証データA_1はメモリ１６に記憶すると共に、通信Ｉ／Ｆ３０に出力され、ネットワーク３２を介してサーバ・コンピュータ４０へ送信される。この送信データは、図４のデータＤｃ１に相当する。

ステップＰ２：サーバ・コンピュータ４０は、クライアント・コンピュータ１０から認証データＡ及び暗証データＣを受信すると共に、乱数Ｑを生成し暗証データＳ、認証データＱを計算しクライアント・コンピュータ１０に送信する。これと共に、記憶されている隠蔽鍵K_0を新規の隠蔽鍵K_1に更新する。このプロセスは、図３のステップ１２０と、図４のプロセスＰｓ１に相当する。

すなわち、サーバ・コンピュータ４０では、通信Ｉ／Ｆ６０を介して検証器５０にクライアント・コンピュータ１０からの暗証データC_1及び認証データA_1が入力される。このとき、サーバ・コンピュータ４０では、乱数発生器４４において乱数Q_1を生成する。生成された乱数Q_1、メモリ４６に記憶されている隠蔽鍵K_0及び隠蔽鍵K_0を構成するC_0，S_0，Q_0，R_0は認証用データ演算器４８へ入力される。また、検証器５０は、クライアント・コンピュータ１０からの暗証データC_1及び認証データA_1を認証用データ演算器４８へ出力する。

認証用データ演算器４８は、その乱数Q_1、受信した暗証データC_1及び記憶されている隠蔽鍵K_0及び隠蔽鍵K_0を構成する認証データQ_0を用いて上記関数ｚ、ｗにより新規の暗証データS_1、及び新規の認証データB_1を求める。この求めた新規の暗証データS_1、及び認証データB_1は通信Ｉ／Ｆ６０に出力され、ネットワーク３２を介してクライアント・コンピュータ１０へ送信される。

この送信データは、図４のデータＤｓ１に相当する。

このとき、サーバ・コンピュータ４０では、初期値としての隠蔽鍵Ｋ。を構成する各データについて新規のデータが入手できている。すなわち、暗証データＣについてはクライアント・コンピュータ１０から受信した暗証データC_1、暗証データＳについては認証用データ演算器４８で計算した暗証データS_1、認証データＱについては乱数発生器４４で発生した乱数Q_1、認証データＲについてはクライアント・コンピュータ１０から受信した認証データＡから逆算すなわち隠蔽鍵K_0を減算することで得られる乱数R_1である。

そこで、これらの暗証データC_1、暗証データS_1、認証データQ_1、認証データR_1を、新規のデータとして更新すると共に、新規の隠蔽鍵K_1として更新する。これによって、サーバ・コンピュータ４０では、隠蔽鍵Ｋの履歴として最新のデータに自動的に更新することができる。

ステップＰ３：クライアント・コンピュータ１０は、サーバ・コンピュータ４０から認証データＢ及び暗証データＳを受信すると共に、乱数Ｒを生成し暗証データC_2、認証データA_2を計算しサーバ・コンピュータ４０に送信する。これと共に、記憶されている隠蔽鍵K_0を新規の隠蔽鍵K_1に更新する。このプロセスは、図３のステップ１３０と、図４のプロセスＰｃ２に相当する。

すなわち、クライアント・コンピュータ１０では、通信Ｉ／Ｆ３０を介して検証器２０にサーバ・コンピュータ４０からの暗証データS_1及び認証データB_1が入力される。このとき、クライアント・コンピュータ１０では、乱数発生器１４において乱数R_2を生成する。生成された乱数Q_2、メモリ４６に記憶されている隠蔽鍵K_0及び隠蔽鍵K_0を構成するC_0，S_0，Q_0，R_0は認証用データ演算器１８へ入力される。また、検証器２０は、サーバ・コンピュータ４０からの暗証データS_1及び認証データB_1を認証用データ演算器１８へ出力する。

このとき、クライアント・コンピュータ１０では、初期値としてメモリ１６に記憶されている隠蔽鍵Ｋ。を構成する各データについて新規のデータ（新規の隠蔽鍵K_1を構成するデータ）が入手できている。すなわち、暗証データＣについてはサーバ・コンピュータ４０から受信した暗証データS_1から逆算すなわち隠

蔽鍵K_0を構成したメモリ16に記憶されている認証データQ_0を減算することで得られる暗証データC_1、またはメモリ16に記憶されている前回送信した暗証データC_1が対応する。暗証データSについてはサーバ・コンピュータ40から受信した暗証データS_1、認証データQについてはサーバ・コンピュータ40から受信した認証データB_1から逆算すなわち隠蔽鍵K_0を減算することで得られる認証データQ_1、認証データRについては前回生成した乱数R_1である。

そこで、これらの暗証データC_1、暗証データS_1、認証データQ_1、認証データR_1を、新規のデータとして更新すると共に、新規の隠蔽鍵K_1として更新する。これによって、クライアント・コンピュータ10では、サーバ・コンピュータ40と同一の隠蔽鍵Kを、最新のデータに自動的に更新することができる。また、認証用データ演算器18は、生成した乱数R_2、更新した履歴データK_1の認証データR_1、受信した暗証データS_1及び新規の隠蔽鍵K_1を用いて上記関数y、vにより新規の暗証データC_2、及び新規の認証データA_2を求める。この求めた新規の暗証データC_2、及び認証データA_2はメモリ16に記憶すると共に、通信I／F30に出力され、ネットワーク32を介してサーバ・コンピュータ40へ送信される。この送信データは、図4のデータDc2に相当する。

ステップP4：上記ステップP2及びP3のプロセスを所定回数mだけ実行する。なお、本実施の形態では、所定回数mは、少なくとも1回のデータ授受を含む。このため、繰り返しを行わない回数（m＝1）を含むものである。すなわち、クライアント・コンピュータ10とサーバ・コンピュータ40との間でなされるデータ授受のときには、既に双方でなされたデータ授受の履歴データが利用されるため、1回のデータ授受であっても、その授受のときにはクライアント・コンピュータ10とサーバ・コンピュータ40との間の履歴を含んでデータ授受がなされるため、単なるデータ授受ではなく、履歴データの授受となるので有効である。上記ステップP2及びP3のプロセスを複数回数だけ繰り返すことは、データの正当性の判断精度向上に有効である。

すなわち、上記処理を繰り返すプロセスは、繰り返す回数すなわち実行回数を複数回予め定めておくことで、隠蔽鍵Kの値が更新されることで変動し、その変

動を第三者が把握することを抑制することが可能になる。このように複数回とすることで、クライアント・コンピュータ１０及びサーバ・コンピュータ４０で共通に保持する隠蔽鍵Ｋは複数回数だけ今までの履歴に従って最新の状態に更新されるので、隠蔽鍵Ｋを導出することが困難になる。

ステップＰ２及びＰ３のプロセスを所定回数ｍだけ実行した結果、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の各々には、隠蔽鍵K_m及び隠蔽鍵K_mを構成するC_m，S_m，Q_m，R_mの値が保持される。なお、ｍ＝１のときは、１回のデータ授受の値が保持される。

なお、処理を繰り返す実行プロセスは、図３のステップ１４０の判断によるプロセス実行と、図４のプロセスＰｃ２からＰｓｍ及びＰｃｍのプロセスについて、プロセスＰｃ１からＰｓ１及びＰｃ２のプロセスを繰り返したことに相当する。

ステップＰ５：上記プロセスが終了した後に、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の各々では、受信したデータの正当性が成立するか否かを検査し、成立すれば相互認証が成功したものとして両者の関係を許諾し、非成立のときは相互認証が不成功であるとして両者の関係を拒否する。このプロセスは、図３のステップ１５０と、図４のプロセスPs_{m+1}及びPc_{m+1}に相当する。

１回の実行の後に認証する場合には、クライアント・コンピュータ１０から１回目のデータ送信がなされるが、そのときクライアント・コンピュータ１０は、クライアント・コンピュータ１０とサーバ・コンピュータ４０との履歴を含む初期値として記憶された隠蔽鍵K_0により生成される、認証データA_1、及び暗証データC_1をサーバ・コンピュータ４０へ送信する。このプロセスは、図４のプロセスＰｃ１の後にデータDc_1を送信することに相当する。

サーバ・コンピュータ４０では、通信Ｉ／Ｆ６０を介して検証器５０にクライアント・コンピュータ１０からの暗証データC_1及び認証データA_1が入力され、暗証データC_1について検証器５０において正当性を検証する。受信した暗証データC_1は、前回の履歴のデータに基づいて生成されているため、サーバ・コンピュータ４０では、最新の状態に更新記憶されている隠蔽鍵K_0（ここでは初期

値）を構成する暗証データS$_0$及び認証データR$_0$を用いて上述の関数yの計算結果と、受信したデータが一致するか否かを判別し、一致する場合は正当性を認め不一致の場合は正当性を否定する。正当性が認められたときには、OK装置52で正当性があることを報知した後に処理を継続し、否認されたときにはNG装置54で不当であることを報知した後に処理を終了する。

正当性が認められて処理が継続されたときには、上記ステップP2と同様にして、乱数発生器44において乱数Q$_1$を生成し、認証用データ演算器48において暗証データS$_1$、認証データB$_1$を生成して、クライアント・コンピュータ10へ送信すると共に、隠蔽鍵を最新の隠蔽鍵K$_1$に更新する。

この認証プロセスは、図4のプロセスPs$_{m+1}$の処理に相当する。この場合、繰り返して実行していないので、m＝0で処理したことに相当する。すなわち、クライアント・コンピュータ10からサーバ・コンピュータ40へデータを送信する毎に、サーバ・コンピュータ40側でクライアント・コンピュータ10から受け取った履歴を含むデータを用いて認証を行うことができる。

一方、クライアント・コンピュータ10では、通信I／F30を介して検証器20にサーバ・コンピュータ40からの暗証データS$_1$及び認証データB$_1$が入力される。クライアント・コンピュータ10では、暗証データS$_1$について検証器20において正当性を検証する。受信した暗証データS$_1$は、暗証データCと同様に前回の履歴のデータに基づいてサーバ・コンピュータ40において生成されているため、クライアント・コンピュータ10では、最新の状態に更新記憶されている隠蔽鍵K$_0$（ここでは初期値）を構成する暗証データC$_0$及び認証データQ$_0$を用いて上述の関数zの計算結果と、受信したデータが一致するか否かを判別し、一致する場合は正当性を認め不一致の場合は正当性を否定する。正当性が認められたときには、OK装置22で正当性があることを報知した後に処理を継続し、否認されたときにはNG装置24で不当であることを報知した後に処理を終了する。

正当性が認められて処理が継続されたときには、クライアント・コンピュータ10とサーバ・コンピュータ40との間で実行すべき処理へと移行する。なお、

クライアント・コンピュータ１０では、サーバ・コンピュータ４０との履歴データKの同一性を維持するため、上記ステップＰ３と同様にして、隠蔽鍵を最新の隠蔽鍵K_1に更新する。

この認証プロセスは、図４のプロセスPc_{m+1}の処理に相当する。この場合、繰り返して実行していないので、ｍ＝０で処理したことに相当する。すなわち、サーバ・コンピュータ４０からクライアント・コンピュータ１０へデータを送信する毎に、クライアント・コンピュータ１０側でサーバ・コンピュータ４０から受け取った履歴を含むデータを用いて認証を行うことができる。

なお、クライアント・コンピュータ１０からサーバ・コンピュータ４０へデータを送信する毎に、またはサーバ・コンピュータ４０からクライアント・コンピュータ１０へデータを送信する毎に、受け取り側で認証を行うことを含めた処理をセッションとして、この認証を含めたセッションを複数回実行してもよい。

次に、複数回の実行を繰り返した後に認証する場合を説明する。この場合には、クライアント・コンピュータ１０からm回目のデータ送信がなされ、クライアント・コンピュータ１０は、m回の繰り返しによって更新された隠蔽鍵K_m、により、サーバ・コンピュータ４０へ認証データA_{m+1}、及び暗証データC_{m+1}を送信する。このプロセスは、図４のプロセスPc_mの後にデータDc_{m+1}を送信することに相当する。

まず、サーバ・コンピュータ４０では、通信Ｉ／Ｆ６０を介して検証器５０にクライアント・コンピュータ１０からの暗証データC_{m+1}及び認証データA_{m+1}が入力される。サーバ・コンピュータ４０では、暗証データC_{m+1}について検証器５０において正当性を検証する。受信した暗証データC_{m+1}は、前回の履歴のデータに基づいて生成されているため、サーバ・コンピュータ４０では、最新の状態に更新記憶されている隠蔽鍵K_mを構成する暗証データS_m及び認証データR_mを用いて上述の関数ｙの計算結果と、受信したデータが一致するか否かを判別し、一致する場合は正当性を認め不一致の場合は正当性を否定する。正当性が認められたときには、ＯＫ装置５２で正当性があることを報知した後に処理を継続し、否認されたときにはＮＧ装置５４で不当であることを報知した後に処理を

終了する。

正当性が認められて処理が継続されたときには、上記ステップＰ２と同様にして、乱数発生器４４において乱数Q_{m+1}を生成し、認証用データ演算器４８において暗証データS_{m+1}、認証データB_{m+1}を生成して、クライアント・コンピュータ１０へ送信すると共に、隠蔽鍵を最新の隠蔽鍵K_{m+1}に更新する。この認証プロセスは、図４のプロセスPs_{m+1}の処理に相当する。

一方、クライアント・コンピュータ１０では、通信Ｉ／Ｆ３０を介して検証器２０にサーバ・コンピュータ４０からの暗証データS_{m+1}及び認証データB_{m+1}が入力される。クライアント・コンピュータ１０では、暗証データS_{m+1}について検証器２０において正当性を検証する。受信した暗証データS_{m+1}は、暗証データＣと同様に前回の履歴のデータに基づいてサーバ・コンピュータ４０において生成されているため、クライアント・コンピュータ１０では、最新の状態に更新記憶されている隠蔽鍵K_mを構成する暗証データC_m及び認証データQ_mを用いて上述の関数ｚの計算結果と、受信したデータが一致するか否かを判別し、一致する場合は正当性を認め不一致の場合は正当性を否定する。正当性が認められたときには、ＯＫ装置２２で正当性があることを報知した後に処理を継続する一方、否認されたときにはＮＧ装置２４で不当であることを報知した後に処理を終了する。

正当性が認められて処理が継続されたときには、クライアント・コンピュータ１０とサーバ・コンピュータ４０との間で実行すべき処理へと移行する。なお、クライアント・コンピュータ１０では、サーバ・コンピュータ４０との履歴データＫの同一性を維持するため、上記ステップＰ３と同様にして、隠蔽鍵を最新の隠蔽鍵K_{m+1}に更新する。この認証プロセスは、図４のプロセスPc_{m+1}の処理に相当する。

このように、本実施の形態では、クライアント・コンピュータ１０とサーバ・コンピュータ４０との間の相互認証をするときに、双方で共通の隠蔽鍵Ｋを有し、その隠蔽鍵Ｋを情報授受毎に更新している。このため、情報授受のときのデータを解析しても、認証用のデータを特定することが困難であり、秘匿性を向上する

ことができ、確実に相互認証が可能となる。

上記では、クライアント・コンピュータ１０とサーバ・コンピュータ４０との間を例にして説明したが、インターネット等の非同期ネットワークにおいては、クライアント・コンピュータ１０に対してサーバ・コンピュータ４０では認証が必要である。この場合には、クライアント・コンピュータ１０のユーザＩＤ毎に処理を分離するようにしてもよい。

上記プロセスは、クライアント・コンピュータ１０及びサーバ・コンピュータ４０の処理プログラムとして記録媒体としてのフレキシブルディスクに実行可能な形式で格納することができる。この場合、各装置に挿抜可能なフレキシブルディスクユニット（ＦＤＵ）を接続して、フレキシブルディスクからＦＤＵを介して記録された処理プログラムを実行すればよい。また、処理プログラムをコンピュータ内のＲＡＭや他の記憶領域（例えばハードディスク装置）にアクセス可能に格納にして（インストール）して実行するようにしてもよい。また、予めＲＯＭに記憶してもよい。また、記録媒体としては、ＣＤ－ＲＯＭ，ＭＤ，ＭＯ，ＤＶＤ等のディスクやＤＡＴ等の磁気テープがあり、これらを用いるときには、対応する装置としてＣＤ－ＲＯＭ装置、ＭＤ装置、ＭＯ装置、ＤＶＤ装置、ＤＡＴ装置等を用いればよい。

以上説明したように本発明の第１の実施の形態によれば、第１認証装置及び第２認証装置の間で相互認証するときに、第１認証装置及び第２認証装置の各々に共通に履歴データを記憶すると共に、履歴データを更新するので、安全かつ簡便に相互認証することができ、例えば、クライアント・コンピュータとサーバ・コンピュータとの間で授受される情報から、クライアント・コンピュータの鍵が漏洩することがなく、確実に認証が行える、という効果がある。

（実施の形態２）

図５は、本発明に係る認証システムの第２の実施の形態を示す概略構成図である。この認証システムは、公衆回線網やインターネット等のネットワーク４０を介して相互に接続されたサーバ（第二装置）１０とクライアント（第一装置）２０とにより概略構成されている。この実施の形態では、種々のサービスを提供す

る複数のサーバＡ、Ｂ、Ｃ、…がサーバ１０に接続され、当該サーバ１０が、サーバＡ、Ｂ、Ｃ、…へのアクセスの可否を決定する認証サーバとして機能するようになっている。

サーバ１０は、図６に示すように、ＣＰＵ１１、ＲＡＭ１２、記憶装置１３、入力装置１４、表示装置１５および通信装置１６等により構成され、各部はバス１７により接続されている。

ＣＰＵ（Central Processing Unit ）１１は、記憶装置１３の記憶領域に格納されている各種処理プログラム、入力装置１４や通信装置１６から入力される各種指示、あるいは指示に対応する各種データ等をＲＡＭ１２に格納し、それら入力指示および各種データに応じてＲＡＭ１２に格納した各種処理プログラムに従って各種処理を実行し、その処理結果をＲＡＭ１２に一時的に記憶するとともに、表示装置１５等に出力する。

このＣＰＵ１１は、当該サーバ１０における受信手段および判定手段を構成しており、クライアントＩＤを引数とする一方向関数（一方向関数Ｆｃ）の関数値であるＨＡＳＨｃ、ワンタイムＩＤ（ＳＩＧＮＡＬ）、ＤＨ公開値g^x（Diffie-Hellman公開値の一方）をクライアント２０から受信した場合（すなわち、クライアント２０からアクセスの要求を受けた場合）に、クライアント２０から受信した受信データと記憶装置１３に記憶されている記憶データを用いてワンタイムＩＤおよびＨＡＳＨｃを演算により求め、この演算結果と、クライアント２０から受信したワンタイムＩＤおよびＨＡＳＨｃとの比較により、クライアント２０の正当性を判定する処理を実行する。

また、ＣＰＵ１１は、当該サーバ１０における送信手段を構成しており、クライアント２０が正当であると判定される場合に、上記受信データおよび上記記憶データを用いて、サーバＩＤを引数とする一方向関数（一方向関数Ｆｓ）の関数値であるＨＡＳＨｓを演算により求め、このＨＡＳＨｓと、記憶装置１３に記憶されているＤＨ公開値g^y（Diffie-Hellman公開値の他方）とをクライアント２０に対して送信する処理を実行する。

なお、上記ワンタイムＩＤ（ＳＩＧＮＡＬ）は、サーバ・クライアント間にお

ける認証において一回限り使用可能な識別情報であり、このワンタイムＩＤを生成する場合には、所定の通信単位毎に変化する暗号化鍵Ｋ（可変共有鍵）を記憶装置１３から読み込んで、この暗号化鍵Ｋを引数とするハッシュ関数（一方向関数）の関数値を求め、この関数値から上記ワンタイムＩＤを生成する。

ＲＡＭ（Random Access Memory）１２は、クライアント２０等との間で送受信されるデータなど、認証に関する各種データを一時記憶する記憶領域や、ＣＰＵ１１の作業領域などを備えている。

記憶装置１３は、プログラムやデータ等が記憶される記憶媒体１３ａを有し、この記憶媒体１３ａは磁気的、光学的記録媒体、若しくは半導体メモリで構成されている。この記憶媒体１３ａは記憶装置１３に固定的に設けたもの、若しくは着脱自在に装着するものであり、ＣＰＵ１１により実行される各種処理プログラムや制御データ等を記憶する記憶領域、認証に関する各種データ（例えば、クライアント２０やＩＤ発行管理サーバ３０（後述）から取得したデータ、認証の処理過程で生成されたデータなど）を格納する記憶領域などを備えている。なお、この記憶媒体１３ａに記憶するプログラムやデータなどは、その一部若しくは全部を他のサーバ等からネットワーク４０を介して受信して記憶する構成とすることも可能である。この記憶媒体１３ａには、サーバＩＤ、ＤＨ公開値g^r、クライアント２０との間で共有化された乱数Ｒなどが、認証処理を開始する前段階で予め格納された状態となっている。

入力装置１４は、キーボードやポインティングデバイス等により構成され、入力指示信号をＣＰＵ１１に対して出力する。

表示装置１５は、ＣＲＴ（Cathode Ray Tube）やＬＣＤ（Liquid Crystal Display）等により構成され、ＣＰＵ１１から入力される表示データを表示する。通信装置１６は、モデムやルータ、ブリッジ等により構成され、ネットワーク４０を介してクライアント２０等より受信したデータをＣＰＵ１１に出力するとともに、ＣＰＵ１１より受信したデータをネットワーク４０を介してクライアント２０等に対して出力する。

一方、クライアント２０は、図７に示すように、ＣＰＵ２１、ＲＡＭ２２、記

憶装置２３、入力装置２４、表示装置２５および通信装置２６等により構成され、各部はバス２７により接続されている。具体的に、クライアント２０としては、例えば、パーソナルコンピュータや、ＰＤＡ（Personal Digital Assistance）等の携帯情報端末、インターネット接続サービスを利用可能な携帯電話などが挙げられる。なお、このクライアント２０の各構成要素は、前述したサーバ１０の各構成要素とほぼ同様であるので、相違点のみを以下に説明する。

すなわち、クライアント２０のＣＰＵ２１は、当該クライアント２０における送信手段を構成しており、入力装置２４からの指示入力等に基づいて、ワンタイムＩＤ（ＳＩＧＮＡＬ）を生成するとともに、クライアントＩＤを引数とする一方向関数（一方向関数Ｆｃ）の関数値であるＨＡＳＨｃを求め、これらワンタイムＩＤおよびＨＡＳＨｃと、記憶装置２３に予め記憶されたＤＨ公開値g^x（Diffie-Hellman公開値の一方）とをサーバ１０に対して送信する処理を実行する。

また、ＣＰＵ２１は、当該クライアント２０における受信手段および判定手段を構成しており、サーバＩＤを引数とする一方向関数（一方向関数Ｆｓ）の関数値であるＨＡＳＨｓと、ＤＨ公開値g^y（Diffie-Hellman公開値の他方）とをサーバ１０から受信した場合（すなわち、サーバ１０によってクライアント２０が正当であると判定された場合）に、サーバ１０から受信した受信データと記憶装置２３に記憶されている記憶データを用いてＨＡＳＨｓを演算により求め、この演算結果と、サーバ１０から受信したＨＡＳＨｓとの比較により、サーバ１０の正当性を判定する処理を実行する。

記憶装置２３は、プログラムやデータ等が記憶される記憶媒体２３ａを有し、この記憶媒体２３ａは、上記ＣＰＵ２１により実行される各種処理プログラムや制御データ等を記憶する記憶領域、認証に関する各種データ（例えば、サーバ１０やＩＤ発行管理サーバ３０（後述）から取得したデータ、認証の処理過程で生成されたデータなど）を格納する記憶領域などを備えている。この記憶媒体２３ａには、クライアントＩＤ、ＤＨ公開値g^x、サーバ１０との間で共有化された乱数Ｒなどが、認証処理を開始する前段階で予め格納された状態となっている。

ＩＤ発行管理サーバ３０は、クライアント・サーバ間で共有化される秘密情報

（例えば、ワンタイムＩＤの初期値を生成するのに用いられる乱数Ｒなど）や、クライアントＩＤ、サーバＩＤなどを発行・管理するためのサーバである。このＩＤ発行管理サーバ３０は、クライアント２０を利用するユーザのＩＤ（例えば、クレジットＮｏ、住基ネットＩＤ、社員Ｎｏ、学生Ｎｏ、特定会員Ｎｏなど）に上記秘密情報やパスワードなどを対応付けた状態で格納するデータベースを有している。また、ＩＤ発行管理サーバ３０は、一定の周期で上記データベース内の秘密情報を更新し、この更新した秘密情報をオンライン（例えば、電子メールなど）またはオフライン（例えば、郵送など）で、クライアント２０とサーバ１０の双方に配布するようになっている。なお、上記秘密情報の発行は、クライアント２０またはサーバ１０からの発行依頼に基づくものであってもよい。

次に、上記構成からなる認証システムによって行われる認証方法の第２の実施の形態について、図８に基づいて説明する。この方法は、ＲＦＣ２４０９において規定されたＩＫＥの方式に、本発明に係るワンタイムＩＤ（ＳＩＧＮＡＬ）を適用したものである。

先ず、ステップＳ１では、ＩＫＥによるＳＡ生成に際してイニシエータとなるクライアント２０が、ワンタイムＩＤ（ＳＩＧＮＡＬ）を生成するとともに、ＨＡＳＨｃを演算により求め、これらワンタイムＩＤおよびＨＡＳＨｃと、記憶装置２３に記憶されたＤＨ公開値g^x とをＳＡの提案とともに、レスポンダとなるサーバ１０に対して送信する処理を実行する。

ここで、ワンタイムＩＤであるＳＩＧＮＡＬは、例えば、ハッシュ関数を用いて、次のように生成される。

$$SIGNAL_1 = R$$
$$SIGNAL_2 = hash(K_1)$$
$$SIGNAL_3 = hash(K_2)$$
$$\cdots$$
$$SIGNAL_n = hash(K_{n-1})$$

・・・・・（式２）

上記ＳＩＧＮＡＬの定義式において、ｈａｓｈはハッシュ関数、ＲはＩＤ発行管理サーバ３０からサーバ１０とクライアント２０の双方に発行されて両者間で共有化された乱数、K_i はｉ番目のセッションで生成されたサーバ・クライアント共有の暗号化鍵（可変共有鍵）である。なお、上記セッションは、ＳＡを確立してから当該ＳＡが無効になるまでの通信単位を示している。

すなわち、上記ＳＩＧＮＡＬの定義式によれば、前回のセッションで生成された上記暗号化鍵Ｋを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのＳＩＧＮＡＬとして用いるようにしている。また、最初のセッションでは、サーバ・クライアント間で予め共有化された乱数ＲをＳＩＧＮＡＬの初期値として用いるようにしている。また、上記暗号化鍵 K_i は、例えば、次式（３）から求められる。

K_i ＝ｐｒｆ（共有鍵、g^{xy}、$SIGNAL_i$）

・・・・・（式３）

この式（３）において、g^{xy}はＤＨ共通鍵であり、共有鍵は、サーバ・クライアント間の任意の共有鍵である。

一方、ＨＡＳＨｃは、次式（４）に示すように、共有鍵、ＤＨ公開値 g^x 、ＩＤｃ（クライアントＩＤ）およびＳＩＧＮＡＬを引数とする疑似乱数関数（鍵付きハッシュ関数）の関数値として求められる。

ＨＡＳＨｃ＝ｐｒｆ（共有鍵、g^x 、ＩＤｃ、ＳＩＧＮＡＬ）

・・・・・（式４）

次いで、ステップＳ２では、サーバ１０が、ＳＩＧＮＡＬとＨＡＳＨｃを演算により求め、これら演算結果と、クライアント２０から受信したＳＩＧＮＡＬおよびＨＡＳＨｃとの比較により、クライアント２０の正当性を判定する処理を実行する。

上記判定の結果、受信データと演算結果とが一致して、クライアント２０が正当であると判定される場合には、ＨＡＳＨｓを演算により求め、このＨＡＳＨｓと、記憶装置１３に記憶されているＤＨ公開値 g^y とを、受諾したＳＡとともにクライアント２０に対して送信する処理を実行する（ステップＳ３）。一方、受

信データと演算結果とが一致せず、クライアント２０が正当でないと判定される場合には、クライアント２０からのアクセスを拒否して、当該認証処理を終了する。

ここで、ＨＡＳＨｓは、次式（５）に示すように、共有鍵、ＤＨ公開値g^x、g^y、ＩＤｓ（サーバＩＤ）およびＳＩＧＮＡＬを引数とする疑似乱数関数（鍵付きハッシュ関数）の関数値として求められる。

ＨＡＳＨｓ＝ｐｒｆ（共有鍵、g^x、g^y、ＩＤｓ、ＳＩＧＮＡＬ）

・・・・・（式５）

また、このステップＳ３においては、記憶装置１３に記憶されているＤＨ公開値ｇｙと、クライアント２０から受信したＤＨ公開値g^xとからＤＨ共通鍵g^{xy}を生成して、ＤＨ共通鍵g^{xy}を記憶装置１３に格納する処理も併せて行う。

次いで、ステップＳ４では、クライアント２０が、ＨＡＳＨｓを演算により求め、この演算結果と、サーバ１０から受信したＨＡＳＨｓとの比較により、サーバ１０の正当性を判定する処理を実行する。

上記判定の結果、受信データと演算結果とが一致して、サーバ１０が正当であると判定される場合には、記憶装置２３に記憶されているＤＨ公開値g^xと、サーバ１０から受信したＤＨ公開値g^yとからＤＨ共通鍵g^{xy}を生成して記憶装置２３に格納した後、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、サーバ１０が正当でないと判定される場合には、サーバ１０へのアクセスを中止して、当該認証処理を終了する。

以上のように、この第２の実施の形態によれば、セッション毎に変化する暗号化鍵Ｋ（可変共有鍵）を引数とするハッシュ関数の関数値をワンタイムＩＤ（ＳＩＧＮＡＬ）として用いるようにしたので、例えば、暗号化鍵Ｋが第三者に漏れたとしても、セッション毎に暗号化鍵Ｋが変化することとなるので、漏れた暗号化鍵Ｋを用いて生成されたワンタイムＩＤ以外のワンタイムＩＤを予測できなくなる。すなわち、盗聴が困難で安全性に優れたワンタイムＩＤを生成することが可能になり、ワンタイムＩＤの将来にわたる安全性（ＰＦＳ）を実現することが可能になる。

また、上記ワンタイムＩＤ（ＳＩＧＮＡＬ）を用いて、クライアント・サーバ間における認証を行うようにしたので、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムＩＤを識別情報として把握することができる。したがって、ＤｏＳ攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ＩＤ情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることもできる。

また、この実施の形態では、クライアント２０の正当性を判定するのに用いる一方向関数Ｆｃとして、共有鍵、ＤＨ公開値g^x、ＩＤｃ（クライアントＩＤ）およびＳＩＧＮＡＬを引数とする疑似乱数関数を用いるとともに、サーバ１０の正当性を判定するのに用いる一方向関数Ｆｓとして、共有鍵、ＤＨ公開値g^x、g^y、ＩＤｓ（サーバＩＤ）およびＳＩＧＮＡＬを引数とする疑似乱数関数を用いるようにしたので、従来の鍵交換・認証方式では３回必要であった通信回数を２回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

（実施の形態３）

前述した第２の実施の形態では、前回のセッションで生成された暗号化鍵（可変共有鍵）を引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのワンタイムＩＤ（ＳＩＧＮＡＬ）として用いるようにしたが、この第３の実施の形態では、前回のセッションで生成された共有鍵と、当該セッションにおける通信順序とを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションの各通信時におけるワンタイムＩＤとして用いるようにしている。この第３の実施の形態特有の部分以外は、第２の実施の形態におけると同様である。この第３の実施の形態において、第２の実施の形態と同一部分には同一符号を付し、その説明を省略する。

図９は本発明に係る認証方法の第３の実施の形態を説明する図である。この第３の実施の形態では、先ず、ステップＰ１において、クライアント２０が、ＳＩＧＮＡＬ$n,_1$（第一のワンタイムＩＤ）を生成するとともに、ＩＤｃ（クライ

アントＩＤ）、ＩＤｓ（サーバＩＤ）、ＤＨ公開値ｇxnおよびＳＩＧＮＡＬn、$_1$を共有鍵K_{n-1}（第一の可変共有鍵）で暗号化し、この暗号化データとＳＩＧＮＡＬn、$_1$とをサーバ１０に対して送信する処理を実行する。

ここで、ＳＩＧＮＡＬは、ｉ番目のセッションにおけるクライアント２０のｊ番目の通信で利用するＳＩＧＮＡＬをＳＩＧＮＡＬi、j、ｉ番目のセッションにおけるサーバ１０のｊ番目の通信で利用するＳＩＧＮＡＬをＳＩＧＮＡＬ’i、jとした場合、次のように生成される。

$SIGNAL_{1,j}$ ＝ｈａｓｈ（Ｒ、ｊ）　　ｉ＝１

$SIGNAL_{i,j}$ ＝ｈａｓｈ（Ki-1、ｊ）　　ｉ≧２

$SIGNAL'_{1,j}$ ＝ｈａｓｈ’（Ｒ、ｊ）　　ｉ＝１

$SIGNAL'_{i,j}$ ＝ｈａｓｈ’（K_{i-1}、ｊ）　　ｉ≧２

・・・・・（式６）

上記ＳＩＧＮＡＬの定義式（６）において、ｈａｓｈとｈａｓｈ’は互いに異なるハッシュ関数、ＲはＩＤ発行管理サーバ３０からサーバ１０とクライアント２０の双方に発行されて両者間で共有化された乱数、Ｋｉ はｉ番目のセッションで共有したＤＨ共通鍵g^{xiyi}（共有鍵）である。

すなわち、上記ＳＩＧＮＡＬの定義式（６）によれば、前回のセッションで生成された共有鍵Ki-1 と今回のセッションにおける通信順序ｊとを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのｊ番目の通信に用いるＳＩＧＮＡＬとしている。ただし、最初のセッション（ｉ＝１）では、サーバ・クライアント間で予め共有化された乱数Ｒと当該セッションにおける通信順序ｊとを引数とするハッシュ関数の関数値を求め、この関数値を最初のセッションのｊ番目の通信に用いるＳＩＧＮＡＬとしている。

次いで、ステップＰ２では、サーバ１０が、ＳＩＧＮＡＬn、$_1$を演算により求め、この演算結果と、クライアント２０から受信したＳＩＧＮＡＬn、$_1$との照合により、クライアント２０を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵K_{j-1}を用いて暗号化データを復号し、この復号したデータに含まれる、ＩＤｃ、ＩＤｓおよびＳＩＧＮＡＬn、$_1$に基づいて、

クライアント２０の正当性を判定する処理を実行する。

上記判定の結果、受信データと、サーバ１０に予め格納された記憶データとが一致して、クライアント２０が正当であると判定される場合には、前述したＳＩＧＮＡＬの定義式に従ってＳＩＧＮＡＬ’$_{n,1}$（第二のワンタイムＩＤ）を生成するとともに、クライアント２０から受信したＤＨ公開値g^{x_n}と当該サーバ１０に予め記憶されたＤＨ公開値g^{y_n}とからＤＨ共通鍵$g^{x_n y_n}$を共有鍵K_n（第二の可変共有鍵）として生成し、この共有鍵K_n、ＩＤｃ、ＩＤｓおよびＳＩＧＮＡＬ’$_{n,1}$を引数とするハッシュ関数ｈの関数値と、ＤＨ公開値g^{y_n}と、ＳＩＧＮＡＬ’$_{n,1}$とをクライアント２０に対して送信する処理を実行する（ステップＰ３）。一方、受信データと記憶データとが一致せず、クライアント２０が正当でないと判定される場合には、クライアント２０からのアクセスを拒否して、当該認証処理を終了する。

次いで、ステップＰ４では、クライアント２０が、ＳＩＧＮＡＬ’$_{n,1}$を演算により求め、この演算結果と、サーバ１０から受信したＳＩＧＮＡＬ’$_{n,1}$との照合により、サーバ１０を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ１０から受信したＤＨ公開値g^{y_n}と当該クライアント２０に予め記憶されたＤＨ公開値g^{x_n}とからＤＨ共通鍵$g^{x_n y_n}$を共有鍵K_nとして生成するとともに、この共有鍵K_nを用いてハッシュ関数ｈの関数値を演算により求め、この演算結果と、サーバ１０から受信したハッシュ関数ｈの関数値との照合により、サーバ１０の正当性を判定する処理を実行する。

上記判定の結果、受信データと演算結果とが一致して、サーバ１０が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、サーバ１０が正当でないと判定される場合には、サーバ１０へのアクセスを中止して、当該認証処理を終了する。

なお、クライアント２０が共有鍵K_iを共有したことをサーバ１０側で確認する必要がある場合には、このステップＰ４でクライアント２０がサーバ１０の正当性を判定した後に、共有鍵K_n、ＩＤｃ、ＩＤｓを引数とするハッシュ関数ｈ

の関数値をサーバ１０に対して送信するようにすればよい。

以上のように、この第３の実施の形態によれば、前回のセッションで生成された共有鍵K_{i-1}（可変共有鍵）と今回のセッションにおける通信順序ｊとを引数とするハッシュ関数の関数値を求め、この関数値を当該セッションのｊ番目の通信にのみ有効なワンタイムＩＤ（ＳＩＧＮＡＬ）として用いるようにしたので、例えば、ｎ番目のセッションで生成した共有鍵K_nが第三者に漏れたとしても、セッション毎に共有鍵K_nが変化することとなるので、漏れた共有鍵K_nを用いて生成されたワンタイムＩＤ（$SIGNAL_{n+1,j}$、$SIGNAL'_{n+1,j}$）以外のワンタイムＩＤを予測できなくなる。すなわち、盗聴が困難で安全性に優れたワンタイムＩＤを生成することが可能になり、ワンタイムＩＤの将来にわたる安全性（ＰＦＳ）を実現することが可能になる。

また、上記ワンタイムＩＤ（ＳＩＧＮＡＬ）を用いて、クライアント・サーバ間における認証を行うようにしたので、前述した第２の実施の形態と同様、大量の計算要求・応答要求などによる計算量やメモリへのＤｏＳ攻撃を防止することができ、オープンなネットワーク環境下においても、ＩＤ情報の保護を図り、通信の安全性を向上させることができる。

なお、ＤｏＳ攻撃を防止する手法の一つとして、クッキー（乱数）を用いた手法が一般に知られている。この方法によれば、ＩＰアドレスとクッキー生成者しか知らない秘密を組み合わせことにより、同一ＩＰアドレスからのＤｏＳ攻撃を防ぐことができる。これに対して、本実施の形態のＳＩＧＮＡＬの場合には、ＤＨ共通鍵を知らない限り、次回有効となるＳＩＧＮＡＬを予測することができない。よって、毎回の通信にＳＩＧＮＡＬを利用することにより、クッキーと同様の効果が得られる。さらに、クッキーの場合はセッション中にＩＰアドレスが変わることを許さないが、ＳＩＧＮＡＬは変わっても良い。また、クッキーを用いた場合ＩＰアドレスを偽造したＤｏＳ攻撃を防ぐことができないが、ワンタイムＩＤではＩＰアドレスが関係ないためこのような攻撃も防ぐことができる。

また、本実施の形態において、例えば、クライアント２０がプロトコルの最初のメッセージを送り（ステップＰ１）、サーバ１０がそれに対応してＤＨ鍵交換

の計算を行い（ステップＰ２）、２番目のメッセージを送った（ステップＰ３）場合を考える。もし、サーバ１０のメッセージが途中で消失、もしくは攻撃者に横取りされ、クライアント２０が受け取ることができなかった場合、クライアント２０はもう一度最初のメッセージを送信する必要がある。このとき、サーバ１０は正しいクライアント２０が通信を送りなおしてきたのか、攻撃者が最初のメッセージを読み取りリプレイ攻撃を行っているのか、判断することができない。そこで、クライアント２０はもう一度最初のメッセージを送りなおす場合、最初のチャレンジの際に送ったメッセージと同一の内容のものを送ることにし、サーバ１０も以前返信したメッセージのコピーをそのまま送ることにする。これにより、無駄なＤＨ鍵交換の計算を避けることができ、リプレイ攻撃によるＤｏＳ攻撃を防ぐことができる。

なお、この実施の形態では、前回のセッションで生成された共有鍵（ＤＨ共通鍵）K_{i-1} と今回のセッションにおける通信順序ｊとを引数とするハッシュ関数の関数値を求め、この関数値を当該セッションのｊ番目の通信にのみ有効なワンタイムＩＤ（ＳＩＧＮＡＬ）として生成するようにしたが、例えば、次のようにＳＩＧＮＡＬを生成することも可能である。

$SS_i = h1(K_{i-1})$

$SIGNAL_{i,j} = hash(SS_i, j)$

$SIGNAL'_{i,j} = hash'(SS_i, j)$

・・・・・（式７）

上記ＳＩＧＮＡＬの定義式（７）において、SS_i は（ｉ－１）番目のセッションで共有したＤＨ共通鍵K_{i-1} を引数とするハッシュ関数の関数値である。

また、この場合には、ｉ番目のセッションで用いられる認証用鍵をＡＫｉ、暗号化鍵をSK_i として、これら鍵を、例えば、

$AK_i = h2(K_{i-1})$、$SK_i = h3(K_{i-1})$

という式から求めるようにしてもよい。なお、ｈ１、ｈ２、ｈ３は、衝突のない一方向性ハッシュ関数である。

このようにSS_i から認証用鍵および暗号化鍵を生成する場合には、前述した

ステップＰ１において、クライアント２０が、ＩＤｃ、ＩＤｓ、ＤＨ公開値g^{x_n}およびＳＩＧＮＡＬ$_{n,1}$を暗号化してサーバ１０に対して送信する際に、認証用鍵ＡＫn を用いるようにする。また、ステップＰ３において、サーバ１０がクライアント２０に対して送信するハッシュ関数ｈに、暗号化鍵ＳＫ$_n$、ＩＤｃ、ＩＤｓおよびＳＩＧＮＡＬ’$_{n,1}$を引数とするハッシュ関数を用いるようにする。

そうすることで、攻撃者が、仮に、ＳＳ$_i$、ＡＫ$_i$、ＳＫ$_i$の何れか１つの値を知ることができたとしても、その他の値を計算することはできない。よって、攻撃者がｉ番目のセッションにおいて正規ユーザに成りすまし、鍵交換を行うためには、ＡＫ$_i$、ＳＩＧＮＡＬ、正規ユーザのＩＤ情報（ＩＤｓ、ＩＤｃ）が必要となり、暗号通信するためには、ＳＫ$_i$ ＳＩＧＮＡＬ、正規ユーザのＩＤ情報、通信回数の情報が必要となる。

また、ｎ番目のセッションにおけるクライアント２０のＤＨ公開値g^{x_n}は、認証鍵ＡＫ$_i$（ｈ２（Ｋ$_{i-1}$））を用いて暗号化される。よって、ＡＫ$_i$を知らない攻撃者はg^{x_n}を知ることができない。そのため、本方式で生成・共有されるDiffie-Hellman共通鍵は計算量的、かつ情報論的に安全である。

（実施の形態４）

前述した第２の実施の形態および　３の実施の形態では、認証と同時にDiffie-Hellman鍵交換を行うようにしたが、この第４の実施の形態では、Diffie-Hellman鍵交換を省略するようにしている。この第４の実施の形態特有の部分以外は、第２の実施の形態におけると同様である。この第４の実施の形態において、第２の実施の形態と同一部分には同一符号を付し、その説明を省略する。

図１０は本発明に係る認証方法の第４の実施の形態を説明する図である。この第４の実施の形態では、先ず、クライアント２０が、乱数Ｒｃ（第一の乱数）を生成するとともに、サーバ１０との間で予め共有化された共有鍵Ｋ１（第一の共有鍵）および乱数Ｒ０（初期乱数）を引数とする疑似乱数関数ｐｒｆ（Ｋ１、Ｒ０）の関数値をＳＩＧＮＡＬ$_{c1}$（第一のワンタイムＩＤ）として求め（ステップＳ１１）、このＳＩＧＮＡＬ$_{c1}$と、共有鍵Ｋ１で暗号化した乱数Ｒｃとをサーバ

１０に対して送信する処理を実行する（ステップＳ１２）。

次いで、サーバ１０が、乱数Ｒｓ（第二の乱数）を生成するとともに、共有鍵Ｋ１で復号した乱数Ｒｃと共有鍵Ｋ１とを引数とする疑似乱数関数ｐｒｆ（Ｋ１、Ｒｃ）の関数値をＳＩＧＮＡＬ$_{s1}$（第二のワンタイムＩＤ）として求め（ステップＳ１３）、このＳＩＧＮＡＬ$_{s1}$と、共有鍵Ｋ１で暗号化した乱数Ｒｓと、乱数Ｒ０＋Ｒｃ（乱数Ｒ０、Ｒｃを引数とする所定の演算結果、例えば、両者の排他的論理和など）とをクライアント２０に対して送信する処理を実行する（ステップＳ１４）。

次いで、クライアント２０が、乱数Ｒｃと共有鍵Ｋ１に基づいてＳＩＧＮＡＬ$_{s1}$を演算により求め、この演算結果とサーバ１０から受信したＳＩＧＮＡＬ$_{s1}$との比較により、サーバ１０を識別するとともに、乱数Ｒ０＋Ｒｃの受信データと演算結果との比較により、サーバ１０の正当性を判定する処理を実行する（ステップＳ１５）。

上記判定の結果、各々の受信データと演算結果とが一致して、サーバ１０が正当であると判定される場合には、クライアント２０が、乱数Ｒｃおよび乱数Ｒｓに基づいて共有鍵Ｋ２（第二の共有鍵）を生成するとともに、この共有鍵Ｋ２、乱数Ｒｓおよび乱数Ｒｃを引数とする疑似乱数関数ｐｒｆ（Ｋ２、Ｒｓ、Ｒｃ）の関数値をＳＩＧＮＡＬ$_{c2}$（第三のワンタイムＩＤ）として求め、このＳＩＧＮＡＬ$_{c2}$と、乱数Ｒｃ＋Ｒｓ（乱数Ｒｃ、Ｒｓを引数とする所定の演算結果）とをサーバ１０に対して送信する処理を実行する（ステップＳ１６）。一方、受信データと演算結果とが一致せず、サーバ１０が正当でないと判定される場合には、サーバ１０へのアクセスを中止して、当該認証処理を終了する。

サーバ１０は、クライアント２０からＳＩＧＮＡＬ$_{c2}$を受信すると、乱数Ｒｃおよび乱数Ｒｓに基づいて共有鍵Ｋ２を生成するとともに、共有鍵Ｋ２、乱数Ｒｓおよび乱数Ｒｃに基づいてＳＩＧＮＡＬ$_{c2}$を演算により求め、この演算結果とクライアント２０から受信したＳＩＧＮＡＬ$_{c2}$との比較により、クライアント２０を識別するとともに、乱数Ｒｃ＋Ｒｓの受信データと演算結果との比較により、クライアント２０の正当性を判定する処理を実行する（ステップＳ１７）。

上記判定の結果、各々の受信データと演算結果とが一致して、クライアント２０が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、クライアント２０が正当でないと判定される場合には、クライアント２０からのアクセスを拒否して、当該認証処理を終了する。

以上のように、この第４の実施の形態によれば、相互認証の過程で生成された乱数と、相互認証の過程で変化する共有鍵Ｋとを引数とする疑似乱数関数ｐｒｆの関数値をワンタイムＩＤとして用いるようにしたので、前述した第２の実施の形態と同様、ワンタイムＩＤの安全性を高めることができ、迅速かつ安全な相互認証を実現することができる。

（実施の形態５）

前述した第４の実施の形態では、ワンタイムＩＤ（ＳＩＧＮＡＬ）の生成に用いる共有鍵を相互認証の過程で変化させるようにしたが、この第５の実施の形態では、上記共有鍵を固定するようにしている。

すなわち、この第５の実施の形態では、図１１に示すように、先ず、クライアント２０が、乱数Ｒｃ（第一の乱数）を生成するとともに、サーバ１０との間で予め共有化された共有鍵Ｋおよび乱数Ｒ０（初期乱数）を引数とする疑似乱数関数ｐｒｆ（Ｋ、Ｒ０）の関数値を$SIGNAL_{c1}$（第一のワンタイムＩＤ）として求め（ステップＳ２１）、この$SIGNAL_{c1}$と、共有鍵Ｋで暗号化した乱数Ｒｃとをサーバ１０に対して送信する処理を実行する（ステップＳ２２）。

次いで、サーバ１０が、乱数Ｒｓ（第二の乱数）を生成するとともに、共有鍵Ｋで復号した乱数Ｒｃと共有鍵Ｋとを引数とする疑似乱数関数ｐｒｆ（Ｋ、Ｒｃ）の関数値を$SIGNAL_{s1}$（第二のワンタイムＩＤ）として求め（ステップＳ２３）、この$SIGNAL_{s1}$と、共有鍵Ｋで暗号化した乱数Ｒｓと、乱数Ｒ０＋Ｒｃ（乱数Ｒ０、Ｒｃを引数とする所定の演算結果）とをクライアント２０に対して送信する処理を実行する（ステップＳ２４）。

次いで、クライアント２０が、乱数Ｒｃおよび共有鍵Ｋに基づいて$SIGNAL_{s1}$を演算により求め、この演算結果とサーバ１０から受信した$SIGNAL_{s1}$

との比較により、サーバ１０を識別するとともに、乱数Ｒ０＋Ｒｃの受信データと演算結果との比較により、サーバ１０の正当性を判定する処理を実行する（ステップＳ２５）。

上記判定の結果、各々の受信データと演算結果とが一致して、サーバ１０が正当であると判定される場合には、クライアント２０が、乱数Ｒｃ、乱数Ｒｓおよび共有鍵Ｋを引数とする疑似乱数関数ｐｒｆ（Ｋ、Ｒｓ、Ｒｃ）の関数値をＳＩＧＮＡＬ$_{c2}$（第三のワンタイムＩＤ）として求め、このＳＩＧＮＡＬ$_{c2}$と、乱数Ｒｃ＋Ｒｓ（乱数Ｒｃ、Ｒｓを引数とする所定の演算結果）とをサーバ１０に対して送信する処理を実行する（ステップＳ２６）。一方、受信データと演算結果とが一致せず、サーバ１０が正当でないと判定される場合には、サーバ１０へのアクセスを中止して、当該認証処理を終了する。

サーバ１０は、クライアント２０からＳＩＧＮＡＬ$_{c2}$を受信すると、乱数Ｒｃ、乱数Ｒｓおよび共有鍵Ｋに基づいてＳＩＧＮＡＬ$_{c2}$を演算により求め、この演算結果とクライアント２０から受信したＳＩＧＮＡＬ$_{c2}$との比較により、クライアント２０を識別するとともに、乱数Ｒｃ＋Ｒｓの受信データと演算結果との比較により、クライアント２０の正当性を判定する処理を実行する（ステップＳ２７）。

上記判定の結果、各々の受信データと演算結果とが一致して、クライアント２０が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、クライアント２０が正当でないと判定される場合には、クライアント２０からのアクセスを拒否して、当該認証処理を終了する。

以上のように、この第５の実施の形態によれば、相互認証の過程で生成された乱数と共有鍵Ｋとを引数とする疑似乱数関数ｐｒｆの関数値をワンタイムＩＤとして用いるようにしたため、例えば、共有鍵Ｋが第三者に漏れたとしても、乱数によって疑似乱数関数ｐｒｆの関数値が相互認証の過程で順次変化することとなるので、相互認証の過程で生成される乱数がわからない限り、ワンタイムＩＤを予測することができない。したがって、前述した第２～第４の実施の形態と同様、

ワンタイムＩＤの安全性を高めることができ、迅速かつ安全な相互認証を実現することができる。

（実施の形態６）

図１２は本発明に係る認証方法の第６の実施の形態を説明する図である。この第６の実施の形態では、先ず、クライアント２０が、乱数R_{ci}（第一の乱数）を生成するとともに、サーバ１０との間で予め共有化された共有鍵K_i、乱数R_{ci-1}（第一の記憶乱数）および乱数R_{si-1}（第二の記憶乱数）を引数とする疑似乱数関数ｐｒｆ（K_i、R_{ci-1}、R_{si-1}）の関数値をＳＩＧＮＡＬ$_{ci}$（第一のワンタイムＩＤ）として求める処理を実行する（ステップＳ３１）。

なお、R_{ci}はｉ番目のセッションでクライアント２０により生成された乱数、R_{si}はｉ番目のセッションでサーバ１０により生成された乱数、K_iはｉ番目のセッションで使用する可変共有鍵をそれぞれ示している。また、前回（ｉ－１番目）のセッションで生成された乱数R_{ci-1}、R_{si-1}は、サーバ１０とクライアント２０の各記憶装置１３、２３の記憶領域に格納されており、これら乱数R_{ci-1}、R_{si-1}に基づいて、共有鍵K_iが生成されるようになっている。

そして、クライアント２０は、ＳＩＧＮＡＬ$_{c1}$を生成した後、ＩＤｃ（クライアントＩＤ）、ＩＤｓ（サーバＩＤ）および乱数R_{ci}を共有鍵K_iで暗号化した暗号化データE_{Ki}（ＩＤｃ、ＩＤｓ、R_{ci}）と、ＳＩＧＮＡＬ$_{c1}$とをサーバ１０に対して送信する処理を実行する（ステップＳ３２）。

サーバ１０は、クライアント２０からＳＩＧＮＡＬ$_{c1}$を受信すると、共有鍵K_i、乱数R_{ci-1}および乱数R_{si-1}に基づいてＳＩＧＮＡＬ$_{c1}$を演算により求め、この演算結果とクライアント２０から受信したＳＩＧＮＡＬ$_{c1}$との比較により、クライアント２０を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵K_iを用いて暗号化データE_{Ki}（ＩＤｃ、ＩＤｓ、。R_{ci}）を復号し、この復号したデータに含まれるＩＤｃおよびＩＤｓに基づいて、クライアント２０の正当性を判定する処理を実行する。

上記判定の結果、受信データと、サーバ１０に予め格納された記憶データとが一致して、クライアント２０が正当であると判定される場合には、乱数R_{si}（第

二の乱数）を生成するとともに、乱数R_{ci}、乱数R_{si-1}および共有鍵K_iを引数とする疑似乱数関数ｐｒｆ（K_i、R_{ci}、R_{si-1}）の関数値をＳＩＧＮＡＬ$_{s1}$（第二のワンタイムＩＤ）として求める。そして、乱数R_{ci-1}、R_{si-1}を格納していた記憶領域に、乱数R_{ci}、R_{si}をそれぞれ格納するとともに、これら乱数R_{ci}、R_{si}に基づき共有鍵K_{i+1}を生成して記憶する処理を実行する（ステップＳ３３）。

次いで、サーバ１０は、ＩＤｃ、ＩＤｓおよび乱数R_{si}を共有鍵K_iで暗号化した暗号化データE_{Ki}（ＩＤｓ、ＩＤｃ、R_{si}）と、ＳＩＧＮＡＬ$_{s1}$とをクライアント２０に対して送信する処理を実行する（ステップＳ３４）。

一方、受信データと記憶データとが一致せず、クライアント２０が正当でないと判定される場合には、クライアント２０からのアクセスを拒否して、当該認証処理を終了する。

クライアント２０は、サーバ１０からＳＩＧＮＡＬ$_{s1}$を受信すると、共有鍵K_i、乱数R_{ci}および乱数R_{si-1}に基づいてＳＩＧＮＡＬ$_{s1}$を演算により求め、この演算結果とクライアント２０から受信したＳＩＧＮＡＬ$_{s1}$との比較により、サーバ１０を識別し、識別できない場合には、通信を拒否する。一方、識別できる場合には、共有鍵K_iを用いて暗号化データE_{Ki}（ＩＤｓ、ＩＤｃ、R_{si}）を復号し、この復号したデータに含まれるＩＤｃおよびＩＤｓに基づいて、サーバ１０の正当性を判定する処理を実行する。サーバ１０を識別できる場合、通信相手を特定できるだけでなく、サーバ１０が乱数R_{ci}を受け取ったことも確認することができる。

そして、上記判定の結果、受信データと、クライアント２０に予め格納された記憶データとが一致して、サーバ１０が正当であると判定される場合には、乱数R_{ci-1}、R_{si-1}を格納していた記憶領域に、乱数R_{ci}、R_{si}をそれぞれ格納して、これら乱数R_{ci}、R_{si}に基づき共有鍵K_{i+1}を生成・記憶した後（ステップＳ３５）、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと記憶データとが一致せず、サーバ１０が正当でないと判定される場合には、サーバ１０からのアクセスを拒否して、当該認証処理を終了する。

以上のように、この第６の実施の形態によれば、前述した第４の実施の形態と同様の作用・効果が得られるのに加えて、ＩＤｃ、ＩＤｓおよび乱数R_{si}を共有鍵K_iで暗号化した暗号化データE_{Ki}（ＩＤｓ、ＩＤｃ、R_{si}）を通信相手に送信するようにしたことにより、例えば、攻撃者によって暗号化データが書き換えられた場合においても、暗号化データに含まれるＩＤ情報（ＩＤｓ、ＩＤｃ）が正しく復号されないために、このデータを受け取ったサーバ１０またはクライアント２０は、送られてきた暗号化データが誤ったものであることを容易に検出でき、乱数を受け取らずに廃棄することが可能となる。また、ＳＩＧＮＡＬ$_{ci}$の値が他の複数のクライアントと重複した場合においても、暗号化データに含まれるＩＤ情報（ＩＤｓ、ＩＤｃ）を参照することによって、通信相手を容易に特定することができる。

さらに、この第６の実施の形態によれば、通信相手がサーバ・クライアントのＩＤ情報（ＩＤｓ、ＩＤｃ）を正しく暗号化できているか否かを確認することによって、通信相手の正当性を判定するようにしたので、前述した第４の実施の形態では３回必要であった通信回数を２回に低減することが可能となり、より効率的な認証が可能となる。

（実施の形態７）

図１３は本発明に係る認証方法の第７の実施の形態を説明する図である。この第７の実施の形態では、先ず、クライアント２０が、乱数R_{ci}（第一の乱数）を生成するとともに、サーバ１０との間で予め共有化された固定共有鍵Ｋ、乱数R_{ci-1}（第一の記憶乱数）および乱数R_{si-1}（第二の記憶乱数）を引数とする疑似乱数関数ｐｒｆ（Ｋ、R_{ci-1}、R_{si-1}）の関数値をＳＩＧＮＡＬ$_{ci}$（第一のワンタイムＩＤ）として求める処理を実行する（ステップＳ４１）。

なお、R_{ci}はｉ番目のセッションでクライアント２０により生成された乱数、R_{si}はｉ番目のセッションでサーバ１０により生成された乱数をそれぞれ示している。また、前回（ｉ－１番目）のセッションで生成された乱数R_{ci-1}、R_{si-1}は、サーバ１０とクライアント２０の各記憶装置１３、２３の記憶領域に格納されている。

そして、クライアント２０は、ＳＩＧＮＡＬ$_{c1}$を生成した後、ＩＤｃ（クライアントＩＤ）、ＩＤｓ（サーバＩＤ）および乱数Ｒ$_{c1}$を共有鍵Ｋで暗号化した暗号化データＥ$_{K}$（ＩＤｃ、ＩＤｓ、Ｒ$_{c1}$）と、ＳＩＧＮＡＬ$_{c1}$とをサーバ１０に対して送信する処理を実行する（ステップＳ４２）。

サーバ１０は、クライアント２０からＳＩＧＮＡＬ$_{c1}$を受信すると、共有鍵Ｋ、乱数Ｒ$_{ci-1}$および乱数Ｒ$_{si-1}$に基づいてＳＩＧＮＡＬ$_{c1}$を演算により求め、この演算結果とクライアント２０から受信したＳＩＧＮＡＬ$_{c1}$との比較により、クライアント２０を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵Ｋを用いて暗号化データＥ$_{K}$（ＩＤｃ、ＩＤｓ、Ｒ$_{ci}$）を復号し、この復号したデータに含まれるＩＤｃおよびＩＤｓに基づいて、クライアント２０の正当性を判定する処理を実行する。

上記判定の結果、受信データと、サーバ１０に予め格納された記憶データとが一致して、クライアント２０が正当であると判定される場合には、乱数Ｒ$_{si}$（第二の乱数）を生成するとともに、乱数Ｒ$_{ci}$、乱数Ｒ$_{si-1}$および共有鍵Ｋを引数とする疑似乱数関数ｐｒｆ（Ｋ、Ｒ$_{ci}$、Ｒ$_{si-1}$）の関数値をＳＩＧＮＡＬ$_{s1}$（第二のワンタイムＩＤ）として求める。そして、乱数Ｒ$_{ci-1}$、Ｒ$_{si-1}$を格納していた記憶領域に、乱数Ｒ$_{ci}$、Ｒ$_{si}$をそれぞれ格納する処理を実行する（ステップＳ４３）。

次いで、サーバ１０は、ＩＤｃ、ＩＤｓおよび乱数Ｒ$_{si}$を共有鍵Ｋで暗号化した暗号化データＥ$_{K}$（ＩＤｓ、ＩＤｃ、Ｒ$_{si}$）と、ＳＩＧＮＡＬ$_{s1}$とをクライアント２０に対して送信する処理を実行する（ステップＳ４４）。

一方、受信データと記憶データとが一致せず、クライアント２０が正当でないと判定される場合には、クライアント２０からのアクセスを拒否して、当該認証処理を終了する。

クライアント２０は、サーバ１０からＳＩＧＮＡＬ$_{s1}$を受信すると、共有鍵Ｋ、乱数Ｒciおよび乱数Ｒ$_{si-1}$に基づいてＳＩＧＮＡＬ$_{s1}$を演算により求め、この演算結果とクライアント２０から受信したＳＩＧＮＡＬ$_{s1}$との比較により、サーバ１０を識別し、識別できない場合には、通信を拒否する。一方、識別できる場合

には、共有鍵Ｋを用いて暗号化データＥＫ（ＩＤｓ、ＩＤｃ、R_{si}）を復号し、この復号したデータに含まれるＩＤｃおよびＩＤｓに基づいて、サーバ１０の正当性を判定する処理を実行する。サーバ１０を識別できる場合、通信相手を特定できるだけでなく、サーバ１０が乱数R_{ci}を受け取ったことも確認することができる。

そして、上記判定の結果、受信データと、クライアント２０に予め格納された記憶データとが一致して、サーバ１０が正当であると判定される場合には、乱数R_{ci-1}、R_{si-1}を格納していた記憶領域に、乱数R_{ci}、R_{si}をそれぞれ格納して、これら乱数R_{ci}、R_{si}に基づき共有鍵Ｋを生成・記憶した後（ステップＳ４５）、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと記憶データとが一致せず、サーバ１０が正当でないと判定される場合には、サーバ１０からのアクセスを拒否して、当該認証処理を終了する。

以上のように、この第７の実施の形態によれば、前述した第５の実施の形態と同様の作用・効果が得られるのに加えて、例えば、攻撃者によって暗号化データが書き換えられた場合においても、データを受け取ったサーバ１０またはクライアント２０は、送られてきた暗号化データが誤ったものであることを容易に検出でき、乱数を受け取らずに廃棄することが可能となる。また、ＳＩＧＮＡＬ$_{ci}$の値が他の複数のクライアントと重複した場合においても、暗号化データに含まれるＩＤ情報（ＩＤｓ、ＩＤｃ）を参照することによって、通信相手を容易に特定することができる。さらに、この第７の実施の形態によれば、前述した第４の実施の形態では３回必要であった通信回数を２回に低減することが可能となり、より効率的な認証が可能となる。

（実施の形態８）

この第８の実施の形態では、ワンタイムＩＤを用いたリプレイ攻撃の防止方法について説明する。リプレイ攻撃とは、過去に正式な通信者が送信したときに有効であった通信情報を攻撃者（第三者）が盗聴し、再利用する攻撃のことである。

先ず、ＯＳＰＡ（Optimal Strong Password Authentication）と呼ばれるパスワードを利用した認証方式（Chun-Li LIN, Hung-Min SUN, Tzonelih HWANG, A

ttacks and Solutions on Strong- Password Authentication, IEICE TRANS. COMMUN., VOL.E84-B, NO.9, September 2001.）について、図１４に基づいて説明する。

当該認証に先立って、クライアント２０には、ハッシュ関数ｈおよびパスワードＰが予め記憶保持されており、サーバ１０には、ハッシュ関数ｈ、セッション回数ｎ、ＩＤｃ（クライアントＩＤ）および検証用情報h^2（Ｐ@ｎ）が予め記憶保持されている。検証用情報h^2（Ｐ@ｎ）は、クライアント２０の正当性を検証するための情報であって、パスワードＰと通信回数ｎの排他的論理和を用いてハッシュ関数ｈにより生成された情報である。なお、h^2（Ｐ@ｎ）は、ハッシュ関数ｈの計算を２回行うこと、つまりｈ（ｈ（Ｐ@ｎ））を示しており、この数式中の@は排他的論理和を示している。

この認証方式では、先ず、クライアント２０がサーバ１０に対してＩＤｃを送信する（ステップＳ５１）。

サーバ１０は、クライアント２０からＩＤｃを受信すると、この受信したＩＤｃと、予め記憶しているＩＤｃとの比較により、クライアント２０を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ１０に対してセッション回数ｎを送信する（ステップＳ５２）。

クライアント２０は、サーバ１０からセッション回数ｎを受信すると、この受信したセッション回数ｎ、予め記憶しているハッシュ関数ｈおよびパスワードＰを用いて、第１～第３の認証用情報Ｃ１、Ｃ２、Ｃ３を生成し（ステップＳ５３）、これらＣ１、Ｃ２、Ｃ３をサーバ１０に対して送信する（ステップＳ５４）。ここで、Ｃ１＝ｈ（Ｐ@ｎ）@h^2（Ｐ@ｎ）、Ｃ２＝h^2（Ｐ@（ｎ＋１））@ｈ（Ｐ@ｎ）、Ｃ３＝h^3（Ｐ@（ｎ＋１））である。

サーバ１０は、クライアント２０からＣ１、Ｃ２、Ｃ３を受信すると、先ず、受信したＣ１≠Ｃ２であることを確認する。これは、Ｃ１＝ｈ（Ｐ@ｎ）@h^2（Ｐ@ｎ）、Ｃ２＝ｈ（Ｐ@ｎ）@h^2（Ｐ@ｎ）、Ｃ３＝h^3（Ｐ@ｎ）と計算して送った場合においても、サーバ１０がクライアント２０を認証してしまい、次の検証用情報として、h^2（Ｐ@（ｎ＋１））ではなくh^2（Ｐ@ｎ）を記憶

してしまう不具合が発生する可能性があることから、このような不具合の発生を防ぐために行われるものである。

次いで、サーバ１０は、Ｃ１、Ｃ２から、ｈ（Ｐ@ｎ）、ｈ²（Ｐ@（ｎ＋１））を演算により求める。すなわち、受信したＣ１と、予め記憶している検証用情報ｈ²（Ｐ@ｎ）との排他的論理和を求めることでｈ（Ｐ@ｎ）を導き出し、このｈ（Ｐ@ｎ）と受信したＣ２との排他的論理和を求めることでｈ²（Ｐ@（ｎ＋１））を導き出す。

次いで、予め記憶しているハッシュ関数ｈを用いて、求めたｈ（Ｐ@ｎ）からｈ（ｈ（Ｐ@ｎ））を計算し、このｈ（ｈ（Ｐ@ｎ））が、予め記憶している検証用情報ｈ²（Ｐ@ｎ）と一致するか否かを検証する。同時に、求めたｈ²（Ｐ@（ｎ＋１））から上記ハッシュ関数ｈを用いてｈ（ｈ²（Ｐ@（ｎ＋１））を計算し、このｈ（ｈ²（Ｐ@（ｎ＋１））が、受信したＣ３と一致するか否かを検証する（ステップＳ５５）。

これら検証の結果、何れもが一致して、クライアント２０が正当であると判定される場合には、検証用情報をｈ²（Ｐ@ｎ）からｈ²（Ｐ@（ｎ＋１））に更新し、セッション回数をｎからｎ＋１に更新した後、クライアント２０からのアクセスを承諾して、当該認証処理を終了する。一方、上記検証の結果、少なくとも何れか一方が一致せず、クライアント２０が正当でないと判定される場合には、クライアント２０からのアクセスを拒否して、当該認証処理を終了する。

上記認証方式によれば、盗聴者に対して安全な認証を行うことができるとともに、セッション毎に検証用情報をｈ²（Ｐ@ｎ）からｈ²（Ｐ@（ｎ＋１））へと更新することができるという利点がある。

しかしながら、上記認証方式にあっては、一度使用された認証情報Ｃ１、Ｃ２、Ｃ３をもう一度利用することによるリプレイ攻撃を防止することができないという問題点があった。

そこで、本発明者等は、このような問題点を解決する認証方式として、次のような認証方式を開発した。

図１５は、本発明に係る認証方法の第８の実施の形態を説明する図である。こ

の図１５に示すように、クライアント２０に、ハッシュ関数ｈおよびパスワードＰが予め記憶保持され、サーバ１０に、ハッシュ関数ｈ、セッション回数ｎ、ＩＤｃおよび検証用情報h^2（Ｐ@ｎ）が予め記憶保持されている場合には、先ず、クライアント２０がサーバ１０に対してＩＤｃを送信する（ステップＳ６１）。

サーバ１０は、クライアント２０からＩＤｃを受信すると、この受信したＩＤｃと、予め記憶しているＩＤｃとの比較により、クライアント２０を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ１０に対してセッション回数ｎを送信する（ステップＳ６２）。

クライアント２０は、サーバ１０からセッション回数ｎを受信すると、この受信したセッション回数ｎ、予め記憶しているハッシュ関数ｈおよびパスワードＰを用いて、第１～第３の認証用情報Ｃ１、Ｃ２、Ｃ３、ＳＩＧＮＡＬ$_n$を生成し（ステップＳ６３）、これらＣ１、Ｃ２、Ｃ３、ＳＩＧＮＡＬ$_n$をサーバ１０に対して送信する（ステップＳ６４）。ここで、Ｃ１＝ｈ（Ｐ@ｎ）@h^2（Ｐ@ｎ）、Ｃ２＝h^2（Ｐ@（ｎ＋１））@ｈ（Ｐ@ｎ）、Ｃ３＝h^3（Ｐ@（ｎ＋１））、ＳＩＧＮＡＬ$_n$＝ｈ（h^2（Ｐ@ｎ）、ｎ）である。即ち、ｎ番目のセッションで使用するワンタイムＩＤであるＳＩＧＮＡＬ$_n$が、検証用情報h^2（Ｐ@ｎ）およびセッション回数ｎを引数とするハッシュ関数ｈの関数値となっている。

サーバ１０は、クライアント２０からＣ１、Ｃ２、Ｃ３、ＳＩＧＮＡＬ$_n$を受信すると、先ず、予め記憶している検証用情報h^2（Ｐ@ｎ）とセッション回数ｎとに基づいてＳＩＧＮＡＬ$_n$を演算により求め、この演算結果とクライアント２０から受信したＳＩＧＮＡＬ$_n$との比較により、クライアント２０を識別し、識別できない場合には、通信を拒否する。識別できる場合には、受信したＣ１≠Ｃ２であることを確認した後、Ｃ１およびＣ２からｈ（Ｐ@ｎ）、h^2（Ｐ@（ｎ＋１））を演算により求める。

次いで、サーバ１０は、予め記憶しているハッシュ関数ｈを用いて、求めたｈ（Ｐ@ｎ）からｈ（ｈ（Ｐ@ｎ））を計算し、このｈ（ｈ（Ｐ@ｎ））が、予め記憶している検証用情報h^2（Ｐ@ｎ）と一致するか否かを検証する。同時に、

求めたh²（P@（n＋1））から上記ハッシュ関数hを用いてh（h²（P@（n＋1））を計算し、このh（h²（P@（n＋1））が、受信したC3と一致するか否かを検証する（ステップS65）。

これら検証の結果、何れもが一致して、クライアント20が正当であると判定される場合には、検証用情報をh²（P@n）からh²（P@（n＋1））に更新し、セッション回数をnからn＋1に更新した後、クライアント20からのアクセスを承諾して、当該認証処理を終了する。一方、上記検証の結果、少なくとも何れか一方が一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

上記認証方式によれば、検証用情報であるh²（P@n）が攻撃者に知られる虞がないので、次のセッションのSIGNALが攻撃者に予測されることはない。しかも、SIGNALは他のセッションで使用できないので、攻撃者によるリプレイ攻撃を効果的に防ぐことができる。

なお、図16に示すように、ハッシュ関数hおよびパスワードPに加えて、予めセッション回数nもクライアント20に記憶保持されている場合には、前述したステップS61、S62の処理を省略することが可能である。したがって、この場合には、ID情報（IDc）の盗聴に対する保護を図りつつも、攻撃者によるリプレイ攻撃を効果的に防ぐことが可能である。

なお、以上の各実施の形態においては、複数の装置間の認証にワンタイムIDを用いるようにしたが、一装置内の複数のアプリケーション間の認証にワンタイムIDを用いることも可能である。また、以上の各実施の形態においては、本発明に係る認証方法をクライアントサーバシステムに適用した場合について例示したが、これに限られるものではなく、例えば、P2P（Peer to Peer）システムに本発明に係る認証方法を適用することも可能である。

また、本発明に係る認証方法をユーザによるアクセス毎に利用することも可能であり、その場合には、ユーザによるパスワードの入力を促して、パスワード、若しくはパスワードから生成した値（ワンタイムパスワードを含む。）をワンタイムIDとともに認証用のデータとして用いることが可能である。

以上説明したように、本発明によれば、盗聴が困難で安全性に優れたワンタイムＩＤを生成することが可能となり、ワンタイムＩＤの将来にわたる安全性（ＰＦＳ）を実現することが可能となる。

また、本発明におけるワンタイムＩＤの生成方法により生成されたワンタイムＩＤを用いて、装置間（クライアント・サーバ間）の認証を行うようにしたので、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムＩＤを識別情報として把握できるようになる。

したがって、ＤｏＳ攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ＩＤ情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることもできる。

また、本発明によれば、従来の鍵交換・認証方式では３回必要であった通信回数を２回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

本発明は、図面に示す好ましい実施の形態に基づいて説明されてきたが、当業者であれば、本発明の思想を逸脱することなく容易に各種の変更、改変し得ることは明らかである。本発明はそのような変更例も含むものである。

請 求 の 範 囲

1． 通信回線を介して接続された第1認証装置と第2認証装置の相互関係を認証する相互認証方法であって、

前記第1認証装置を特定するための記憶データと、第2認証装置を特定するための記憶データとを、前記第1認証装置及び第2認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを用いて更新した更新結果を履歴データとして、前記第1認証装置及び第2認証装置の各々に共通に記憶する記憶工程と、

前記第1認証装置は、記憶されている履歴データを用いて記憶データを新規に生成しかつ生成した新規の記憶データを前記履歴データを用いて暗号化して第2認証装置に送信する第1送信工程と、前記第2認証装置からの記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第1更新工程と、を含み、

前記第2認証装置は、前記第1認証装置からの記憶データ及び記憶されている履歴データを用いて新規に記憶データを生成しかつ生成した新規の記憶データを前記履歴データを用いて暗号化して第1認証装置に送信する第2送信工程と、前記第1認証装置からの記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第2更新工程とを含み、

前記第1認証装置及び第2認証装置の少なくとも一方の装置において、履歴データに基づいて記憶データの正当性が成立したときに、第1認証装置と第2認証装置の相互関係が正当であると検証することを特徴とする相互認証方法。

2． 前記履歴データを履歴データKとして、該履歴データKとして記憶する、前記第1認証装置を特定するための記憶データは、暗証データC及び認証データRであり、前記第2認証装置を特定するための記憶データは、暗証データS及び認証データQであることを特徴とする請求項1に記載の相互認証方法。

3． 前記第1送信工程は、記憶されている履歴データKの暗証データS及び認証データRを用いて暗証データCを新規に生成しかつ、記憶されている履歴データKの認証データRについて新規に生成し、生成した新規の認証デー

タＲを前記履歴データＫを用いて暗号化して認証データＡを求め、前記認証データＡ及び新規の暗証データＣを第２認証装置に送信し、

前記第１更新工程は、前記第２認証装置からのデータを受信し、前記送信した新規の暗証データＣ、受信した新規に生成された暗証データＳ、受信した新規に生成された認証データＱ、及び前記送信した新規の認証データＲにより、前記履歴データＫを更新し、

前記第２送信工程は、前記第１認証装置からのデータを受信し、受信した新規の暗証データＣ及び記憶されている履歴データＫの認証データＱを用いて暗証データＳを新規に生成しかつ記憶されている履歴データＫの認証データＱについて新規に生成し、生成した新規の認証データＱを記憶した履歴データＫを用いて暗号化して認証データＢを求め、前記認証データＢ及び新規の暗証データＳを第１認証装置に送信し、

前記第２更新工程は、受信した新規の暗証データＣ、新規に生成した暗証データＳ、新規に生成した認証データＱ、及び受信した新規の認証データＲにより、前記履歴データＫを更新し、

前記第１認証装置及び第２認証装置の少なくとも一方の装置において、履歴データＫに基づいて暗証データの正当性が成立したときに、第１認証装置と第２認証装置の相互関係が正当であると検証することを特徴とする請求項２に記載の相互認証方法。

４．　前記記憶工程は、前記第１送信工程、第１更新工程、第２送信工程、及び第２更新工程における認証による更新結果を履歴データとして記憶することを特徴とする請求項１に記載の相互認証方法。

５．　前記認証データＲ及び認証データＱの少なくとも一方は、乱数発生手段により発生された乱数、データ容量、時間データの少なくとも１つであることを特徴とする請求項２に記載の相互認証方法。

６．　前記第１認証装置の第１送信工程では、前記暗証データＳ及び認証データＲによる予め定めた関数の演算結果の値を暗証データＣとして生成し、前記第２認証装置の第２送信工程では、前記暗証データＣ及び前記認証データＱ

による予め定めた関数の演算結果の値を暗証データＳとして生成することを特徴とする請求項２に記載の相互認証方法。

７．　前記第１認証装置の第１送信工程では、前記生成した新規の認証データＲ及び前記履歴データＫによる予め定めた関数の演算結果の値を認証データＡとして求め、前記第２認証装置の第２送信工程では、前記生成した新規の認証データＱ及び前記履歴データＫによる予め定めた関数の演算結果の値を認証データＢとして求めることを特徴とする請求項２に記載の相互認証方法。

８．　前記第１認証装置の検証工程は、前記履歴データＫのうち記憶されている認証データＱ及び前回送信する前に生成した暗証データＣによる予め定めた関数の演算結果の値が受信した暗証データＳと一致するときに前記相互関係が正当であると検証することを特徴とする請求項２に記載の相互認証方法。

９．　前記第２認証装置の検証工程は、前記履歴データＫのうち記憶されている暗証データＳ及び認証データＲによる予め定めた関数の演算結果の値が受信した暗証データＣと一致するときに前記相互関係が正当であると検証することを特徴とする請求項２に記載の相互認証方法。

１０．　前記記憶工程は、前記第１送信工程、第２送信工程、第１更新工程及び第２更新工程を複数実施した結果、得られるデータを履歴データＫとして記憶することを特徴とする請求項２に記載の相互認証方法。

１１．　通信回線を介して接続された第１認証装置と第２認証装置とから成り、前記第１認証装置と第２認証装置との間の相互関係を認証する相互認証装置であって、

前記第１認証装置に設けられ当該第１認証装置を特定するための記憶データを格納する第１のメモリと、

前記第２認証装置に設けられ当該第２認証装置を特定するための記憶データを格納する第２のメモリと、

前記第１認証装置及び第２認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを格納する認証データ記憶手段と、

前記認証データを用いて更新した更新結果を履歴データとして、前記第１認証

装置及び第２認証装置の各々に共通に記憶する履歴データ記憶手段と、

前記第１認証装置又は第２認証装置のうち、認証用データ送信側の認証装置に設けられ、前記履歴データを用いて記憶データを新規に生成する記憶データ生成手段と、

生成した新規の記憶データを前記履歴データを用いて暗号化して認証用データ受信側の認証装置に送信する第１の送信手段と、

認証用データ受信側の認証装置に設けられ、前記認証用データ送信側の認証装置からの記憶データ及び記憶されている履歴データを用いて新規に記憶データを生成する記憶データ生成手段と、

生成した新規の記憶データを前記履歴データを用いて暗号化して前記認証用データ送信側の認証装置に返信する第２の送信手段と、

認証用データ送信側の認証装置に設けられ、前記認証用データ受信側の認証装置から返信された記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第１の更新手段と、

認証用データ受信側の認証装置に設けられ、前記認証用データ送信側の認証装置からの記憶データ及び前記返信した新規の記憶データによって前記履歴データを更新する第２の更新手段とを含み、

前記第１認証装置及び第２認証装置の少なくとも一方の装置において、前記履歴データに基づいて記憶データの正当性が成立したときに、第１認証装置と第２認証装置の相互関係が正当であると検証する検証手段とを備えた、

ことを特徴とする相互認証装置。

１２．　生成した新規の記憶データを前記履歴データを用いて暗号化するための認証用データを演算する演算手段を有することを特徴とする請求項１１記載の相互認証装置。

１３．　前記演算手段により認証用データを生成するに際し、暗号化用のデータを生成する乱数発生手段を有することを特徴とする請求項１２記載の相互認証装置。

１４．複数の相互認証を行う装置間またはアプリケーション間における認証に

おいて、一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするワンタイムＩＤの生成方法。

１５．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするワンタイムＩＤの生成方法。

１６．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするワンタイムＩＤの生成方法。

１７．　一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムＩＤを送信して、他方の装置が、一方の装置から受信したワンタイムＩＤと自らが生成したワンタイムＩＤとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムＩＤを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化す

る可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするワンタイムＩＤの生成方法。

１８．　一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムＩＤを送信して、他方の装置が、一方の装置から受信したワンタイムＩＤと自らが生成したワンタイムＩＤとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムＩＤを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするワンタイムＩＤの生成方法。

１９．　一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、当該ワンタイムＩＤを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムＩＤを送信して、他方の装置が、一方の装置から受信したワンタイムＩＤと自らが生成したワンタイムＩＤとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムＩＤを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成するようにしたことを特徴とするワンタイムＩＤの生成方法。

２０．　装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイム

ＩＤを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、上記第二装置との間で予め共有化された可変共有鍵を用いて上記ワンタイムＩＤを生成するとともに、この生成したワンタイムＩＤと、当該第一装置に予め設定されたＩＤを少なくとも引数とする一方向関数Ｆｃの関数値と、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方とを上記第二装置に対して送信するステップと、

上記第二装置が、上記ワンタイムＩＤおよび上記一方向関数Ｆｃの関数値を演算により求め、この演算結果と、上記第一装置から受信したワンタイムＩＤおよび一方向関数Ｆｃの関数値との照合により、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、当該第二装置に予め設定されたＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とを上記第一装置に対して送信するステップと、

上記第一装置が、上記一方向関数Ｆｓの関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数Ｆｓの関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とする認証方法。

２１．　上記一方向関数Ｆｃとして、所定の共有鍵、上記Diffie-Hellman公開値の一方、上記第一装置に予め設定されたＩＤ、上記ワンタイムＩＤを引数とする疑似乱数関数を用いるとともに、

上記一方向関数Ｆｓとして、上記所定の共有鍵、上記Diffie-Hellman公開値の一方、上記Diffie-Hellman公開値の他方、上記第二装置に予め設定されたＩＤ、上記ワンタイムＩＤを引数とする疑似乱数関数を用いるようにしたことを特徴とする請求項２０に記載の認証方法。

２２．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する

情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、上記第二装置との間で予め共有化された第一の可変共有鍵と当該第一装置の通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムＩＤとして生成するとともに、上記第一の可変共有鍵を用いて、当該第一装置に予め設定されたＩＤ、上記第二装置に予め設定されたＩＤ、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムＩＤを暗号化し、この暗号化データと上記第一のワンタイムＩＤとを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムＩＤとの照合により、上記第一装置を識別するステップと、

上記第二装置が、上記第一装置を識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたＩＤ、当該第二装置に予め設定されたＩＤおよび上記第一のワンタイムＩＤに基づいて、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、上記第一の可変共有鍵と当該第二装置の通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして生成するとともに、上記第一装置から受信したDiffie-Hellman公開値の一方と当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記第一装置に予め設定されたＩＤ、当該第二装置に予め設定されたＩＤおよび上記第二のワンタイムＩＤを引数とする一方向関数ｈの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムＩＤとを上記第一装置に対して送信するステップと、

上記第一装置が、上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムＩＤとの照合により、上記

第二装置を識別するステップと、

上記第一装置が、上記第二装置を識別できた場合に、上記第二装置から受信した上記Diffie-Hellman公開値の他方と当該第一装置に予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数ｈの関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数ｈの関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とする認証方法。

２３．　上記第二のワンタイムＩＤを生成する一方向関数として、上記第一のワンタイムＩＤを生成する一方向関数とは異なる一方向関数を用いるようにしたことを特徴とする請求項２２に記載の認証方法。

２４．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、この第一のワンタイムＩＤと上記第一の乱数とを上記第二装置に対して送信するステップと、

上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、この第二のワンタイムＩＤと上記第二の乱数とを上記第一装置に対して送信するステップと、

上記第一装置が、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記第二装置から受信した上記第二のワンタイムＩＤとの比較により、上記第二装置の正当性を判定するス

テップと、

上記第一装置が、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムＩＤとして求め、この第三のワンタイムＩＤを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記第一装置から受信した上記第三のワンタイムＩＤとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とする認証方法。

２５．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、この第一のワンタイムＩＤと上記第一の乱数を上記第二装置に対して送信するステップと、

上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、この第二のワンタイムＩＤと上記第二の乱数を上記第一装置に対して送信するステップと、

上記第一装置が、上記第一の乱数および上記共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記第二装置から受信した上記第二のワンタイムＩＤとの比較により、上記第二装置の正当性を判定するステップと、

上記第一装置が、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワンタイムＩＤとして求め、この第三のワンタイムＩＤを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記第一装置から受信した上記第三のワンタイムＩＤとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とする認証方法。

２６．　上記第一の乱数と上記第二の乱数を、上記第一装置と上記第二装置との間で予め共有化された共有鍵で暗号化した状態で、送信するようにしたことを特徴とする請求項２４に記載の認証方法。

２７．　上記第一の乱数と上記第二の乱数を、上記第一装置と上記第二装置との間で予め共有化された共有鍵で暗号化した状態で、送信するようにしたことを特徴とする請求項に記載の認証方法。

２８．　上記第二装置が上記第二のワンタイムＩＤと上記第二の乱数とを上記第一装置に対して送信するステップにおいて、上記第二装置は、上記第一装置との間で予め共有化された乱数を初期乱数として、この初期乱数と上記第一の乱数を引数とする所定の演算を行い、この演算結果を上記第一装置に対して送信する一方、上記第一装置は、上記第二装置の正当性の判定材料として、上記第二装置から受信した上記演算結果を、上記第二のワンタイムＩＤとともに用いることを特徴とする請求項２４乃至請求項２６の何れかに記載の認証方法。

２９．　上記第一装置が上記第三のワンタイムＩＤを上記第二装置に対して送信するステップにおいて、上記第一装置は、上記第一の乱数と上記第二の乱数を引数とする所定の演算を行い、この演算結果を上記第二装置に対して送信する一方、上記第二装置は、上記第一装置の正当性の判定材料として、上記第一装置から受信した上記演算結果を、上記第三のワンタイムＩＤとともに用いることを特徴とする請求項２４に記載の認証方法。

３０．　上記第一装置が上記第三のワンタイムＩＤを上記第二装置に対して送信するステップにおいて、上記第一装置は、上記第一の乱数と上記第二の乱数

を引数とする所定の演算を行い、この演算結果を上記第二装置に対して送信する一方、上記第二装置は、上記第一装置の正当性の判定材料として、上記第一装置から受信した上記演算結果を、上記第三のワンタイムＩＤとともに用いることを特徴とする請求項２５に記載の認証方法。

３１．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、当該第一装置に予め設定されたＩＤ、上記第二装置に予め設定されたＩＤおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムＩＤとを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムＩＤとの照合により、上記第一装置を識別するステップと、

上記第二装置が、上記第一装置を識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたＩＤおよび当該第二装置に予め設定されたＩＤに基づいて、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、上記第一装置に予め設定されたＩＤ、当該第二装置に予め設定されたＩＤおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムＩＤ

とを上記第一装置に対して送信するステップと、

上記第二装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップと、

上記第一装置が、上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムＩＤとの照合により、上記第二装置を識別するステップと、

上記第一装置が、上記第二装置を識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる、上記第二装置に予め設定されたＩＤおよび当該第一装置に予め設定されたＩＤに基づいて、上記第二装置の正当性を判定するステップと、

上記第一装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップとを有することを特徴とする認証方法。

３２．　上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後に、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようにしたことを特徴とする請求項３１に記載の認証方法。

３３．　装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で認証を行うサーバであって、

上記クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムＩＤとを上記クライアントから受信する受信手段と、

上記一方向関数の関数値Ｆｃおよび上記ワンタイムＩＤを演算により求め、こ

の演算結果と、上記クライアントから受信した上記ワンタイムＩＤおよび上記一方向関数Ｆｃの関数値との比較により、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、当該サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、当該サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する送信手段とを備えることを特徴とするサーバ。

３４．　装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてサーバとの間で認証を行うクライアントであって、

上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムＩＤを生成するとともに、当該クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値を演算により求め、これらワンタイムＩＤおよび一方向関数Ｆｃの関数値と、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する送信手段と、

上記サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する受信手段と、

上記一方向関数Ｆｓの関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数Ｆｓの関数値との比較により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするクライアント。

３５．　サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通

信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で認証を行い、

上記サーバは、上記クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムＩＤとを上記クライアントから受信する受信手段と、

上記一方向関数の関数値Ｆｃおよび上記ワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムＩＤおよび上記一方向関数Ｆｃの関数値との比較により、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、当該サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、当該サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する送信手段とを備え、また、

上記クライアントは、上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムＩＤを生成するとともに、当該クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値を演算により求め、これらワンタイムＩＤおよび一方向関数Ｆｃの関数値と、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する送信手段と、

上記サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する受信手段と、

上記一方向関数Ｆｓの関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数Ｆｓの関数値との比較により、上記サーバの正当性を判定する判定手段とを備えてなることを特徴とする認証システム。

３６．　装置間またはアプリケーション間における認証において、一回限り

使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤに基づいてクライアントとの間で認証を行うサーバに実行させるプログラムであって、

上記クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムＩＤとを上記クライアントから受信する処理と、

上記一方向関数の関数値Ｆｃおよび上記ワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムＩＤおよび上記一方向関数Ｆｃの関数値との比較により、上記クライアントの正当性を判定する処理と、

上記クライアントが正当であると判定された場合に、上記サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する処理とを上記サーバに実行させることを特徴とするプログラム。

３７．　装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤに基づいてサーバとの間で認証を行うクライアントに実行させるプログラムであって、

上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムＩＤを生成するとともに、上記クライアントに予め設定されたクライアントＩＤを少なくとも引数とする一方向関数Ｆｃの関数値を演算により求め、これらワンタイムＩＤおよび一方向関数Ｆｃの関数値と、上記クライアントに予め記憶されたDi

ffie-Hellman公開値の一方とを上記サーバに対して送信する処理と、

上記サーバに予め設定されたサーバＩＤを少なくとも引数とする一方向関数Ｆｓの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する処理と、

上記一方向関数Ｆｓの関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数Ｆｓの関数値との比較により、上記サーバの正当性を判定する処理とを上記クライアントに実行させることを特徴とするプログラム。

３８．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で認証を行うサーバであって、

上記クライアントとの間で予め共有化された第一の可変共有鍵と上記クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムＩＤとして、この第一のワンタイムＩＤ、上記クライアントに予め設定されたクライアントＩＤ、当該サーバに予め設定されたサーバＩＤ、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方を上記第一の可変共有鍵で暗号化した暗号化データと、上記第一のワンタイムＩＤとを上記クライアントから受信する受信手段と、

上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムＩＤとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記クライアントＩＤ、上記サーバＩＤおよび上記第一のワンタイムＩＤに基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、上記第一の

可変共有鍵と当該サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして生成するとともに、上記クライアントから受信したDiffie-Hellman公開値の一方と当該サーバに予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記クライアントＩＤ、上記サーバＩＤおよび上記第二のワンタイムＩＤを引数とする一方向関数ｈの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムＩＤとを上記クライアントに対して送信する送信手段とを備えることを特徴とするサーバ。

３９．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてサーバとの間で認証を行うクライアントであって、　上記サーバとの間で予め共有化された第一の可変共有鍵と当該クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムＩＤとして生成するとともに、上記第一の可変共有鍵を用いて、当該クライアントに予め設定されたクライアントＩＤ、上記サーバに予め設定されたサーバＩＤ、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムＩＤを暗号化し、この暗号化データと上記第一のワンタイムＩＤとを上記サーバに対して送信する送信手段と、

上記第一の可変共有鍵と上記サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムＩＤとし、Diffie-Hellman共通鍵を第二の可変共有鍵として、上記第二のワンタイムＩＤ、上記第二の可変共有鍵、上記クライアントＩＤおよび上記サーバＩＤを引数とする一方向関数ｈの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方と、上記第二のワンタイムＩＤとを上記サーバから受信する受信手段と、

上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記サーバか

ら受信した上記第二のワンタイムＩＤとの照合により、上記サーバを識別し、上記サーバを識別した場合に、上記サーバから受信した上記Diffie-Hellman公開値の他方と当該クライアントに予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数ｈの関数値を演算により求め、この演算結果と、上記サーバから受信した一方向関数ｈの関数値との照合により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするクライアント。

４０．　サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で認証を行い、

上記サーバは、上記クライアントとの間で予め共有化された第一の可変共有鍵と上記クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムＩＤとして、この第一のワンタイムＩＤ、上記クライアントに予め設定されたクライアントＩＤ、当該サーバに予め設定されたサーバＩＤ、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方を上記第一の可変共有鍵で暗号化した暗号化データと、上記第一のワンタイムＩＤとを上記クライアントから受信する受信手段と、

上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムＩＤとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記クライアントＩＤ、上記サーバＩＤおよび上記第一のワンタイムＩＤに基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、上記第一の可変共有鍵と当該サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして生成するとともに、上記クライアントから受信したDiffie-Hellman公開値の一方と当該サーバに予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記クライアントＩＤ、上記サーバＩＤおよび上記第二のワンタイムＩＤを引数とする一方向関数ｈの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムＩＤとを上記クライアントに対して送信する送信手段とを備え、また、

上記クライアントは、上記サーバとの間で予め共有化された第一の可変共有鍵と当該クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムＩＤとして生成するとともに、上記第一の可変共有鍵を用いて、当該クライアントに予め設定されたクライアントＩＤ、上記サーバに予め設定されたサーバＩＤ、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムＩＤを暗号化し、この暗号化データと上記第一のワンタイムＩＤとを上記サーバに対して送信する送信手段と、

上記第一の可変共有鍵と上記サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムＩＤとし、Diffie-Hellman共通鍵を第二の可変共有鍵として、上記第二のワンタイムＩＤ、上記第二の可変共有鍵、上記クライアントＩＤおよび上記サーバＩＤを引数とする一方向関数ｈの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方と、上記第二のワンタイムＩＤとを上記サーバから受信する受信手段と、

上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムＩＤとの照合により、上記サーバを識別し、上記サーバを識別した場合に、上記サーバから受信した上記Diffie-Hellman公開値の他方と当該クライアントに予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数ｈの関数値を演算により求め、この

演算結果と、上記サーバから受信した一方向関数ｈの関数値との照合により、上記サーバの正当性を判定する判定手段とを備えることを特徴とする認証システム。

４１．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で相互に認証を行うサーバであって、

上記クライアントとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして、この第一のワンタイムＩＤと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、この第二のワンタイムＩＤと上記第二の乱数とを上記クライアントに対して送信する送信手段と、

上記第一の乱数、上記第二の乱数および第二の共有鍵を引数とする一方向関数の関数値を第三のワンタイムＩＤとして、この第三のワンタイムＩＤを上記クライアントから受信する第二受信手段と、

上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムＩＤとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするサーバ。

４２．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイム

ＩＤを用いてサーバとの間で相互に認証を行うクライアントであって、

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、この第一のワンタイムＩＤと上記第一の乱数とを上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして、この第二のワンタイムＩＤと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムＩＤとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムＩＤとして求め、この第三のワンタイムＩＤを上記サーバに対して送信する第二送信手段とを備えることを特徴とするクライアント。

４３．　サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で相互に認証を行うように構成され、

上記サーバは、上記クライアントとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして、この第一のワンタイムＩＤと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、この第二のワンタイムＩＤと上記第二の乱数とを上記クライアントに対して送信する送信手段と、

上記第一の乱数、上記第二の乱数および第二の共有鍵を引数とする一方向関数の関数値を第三のワンタイムＩＤとして、この第三のワンタイムＩＤを上記クライアントから受信する第二受信手段と、

上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムＩＤとの比較により、上記クライアントの正当性を判定する判定手段とを備え、また、

上記クライアントは、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、この第一のワンタイムＩＤと上記第一の乱数とを上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして、この第二のワンタイムＩＤと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムＩＤとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムＩＤとして求め、この第三のワンタイムＩＤを上記サーバに対して送信する第二送信手段とを備えることを特徴とする認証システム。

４４．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはア

プリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で相互に認証を行うサーバであって、

上記クライアントとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして、この第一のワンタイムＩＤと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、この第二のワンタイムＩＤと上記第二の乱数を上記クライアントに対して送信する送信手段と、

上記共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムＩＤとして、この第三のワンタイムＩＤを上記クライアントから受信する第二受信手段と、

上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムＩＤとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするサーバ。

４５．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてサーバとの間で相互に認証を行うクライアントであって、

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、この第一のワンタイムＩＤと上記第一の乱数を上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして、この第二のワンタイムＩＤと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムＩＤとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワンタイムＩＤとして求め、この第三のワンタイムＩＤを上記サーバに対して送信する第二送信手段とを備えることを特徴とするクライアント。

４６．　サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で相互に認証を行うように構成され、

上記サーバは、上記クライアントとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして、この第一のワンタイムＩＤと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、この第二のワンタイムＩＤと上記第二の乱数を上記クライアントに対して送信する送信手段と、

上記共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムＩＤとして、この第三のワンタイムＩＤを上記クライアントから受信する第二受信手段と、

上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワンタイムＩＤを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムＩＤとの比較により、上記クライアントの正当性を判定する判定手段とを備え、また、

上記クライアントは、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、この第一のワンタイムＩＤと上記第一の乱数を上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムＩＤとして、この第二のワンタイムＩＤと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記共有鍵に基づいて上記第二のワンタイムＩＤを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムＩＤとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワンタイムＩＤとして求め、この第三のワンタイムＩＤを上記サーバに対して送信する第二送信手段とを備えることを特徴とする認証システム。

４７．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で相互に認証を行うサーバであって、

上記クライアントとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムＩＤとして、この第一のワンタイムＩＤを上記クライアントから受信するとともに、上記クライアントで生成された第一の乱数、上記クライアントに予め設定されたクライア

ントＩＤ、当該サーバに予め設定されたサーバＩＤを上記共有鍵で暗号化した第一の暗号化データを上記クライアントから受信する受信手段と、

上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムＩＤとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる上記クライアントＩＤおよび上記サーバＩＤに基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、上記クライアントＩＤ、上記サーバＩＤおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムＩＤとを上記クライアントに対して送信する送信手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするサーバ。

４８．　複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてサーバとの間で相互に認証を行うクライアントであって、

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、当該クライアントに予め設定されたクライアントＩＤ、上記サーバに予め設定されたサーバＩＤおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムＩＤとを上記サーバに対して送信する送信手段と、

上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムＩＤとして、この第二のワンタイムＩＤを上記サーバから受信するとともに、上記サーバで生成された第二の乱数、上記クライアントＩＤおよび上記サーバＩＤを上記共有鍵で暗号化した第二の暗号化データを上記サーバから受信する受信手段と、

上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムＩＤとの照合により、上記サーバを識別し、上記サーバを識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる上記サーバＩＤおよび上記クライアントＩＤに基づいて、上記サーバの正当性を判定する判定手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするクライアント。

４９．　サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムＩＤとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムＩＤを生成し、この生成されたワンタイムＩＤを用いてクライアントとの間で相互に認証を行うように構成され、

上記サーバは、上記クライアントとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムＩＤとして、この第一のワンタイムＩＤを上記クライアントから受信するとともに、上記クライアントで生成された第一の乱数、上記クライアントに予め設定されたクライアントＩＤ、当該サーバに予め設定されたサーバＩＤを上記共有鍵で暗号化した第一の暗号化データを上記クライアントから受信する受信手段と、

上記第一のワンタイムＩＤを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムＩＤとの照合により、上記クライアント

を識別し、上記クライアントを識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる上記クライアントＩＤおよび上記サーバＩＤに基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムＩＤとして求め、上記クライアントＩＤ、上記サーバＩＤおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムＩＤとを上記クライアントに対して送信する送信手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備え、また、

上記クライアントは、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムＩＤとして求め、当該クライアントに予め設定されたクライアントＩＤ、上記サーバに予め設定されたサーバＩＤおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムＩＤとを上記サーバに対して送信する送信手段と、

上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムＩＤとして、この第二のワンタイムＩＤを上記サーバから受信するとともに、上記サーバで生成された第二の乱数、上記クライアントＩＤおよび上記サーバＩＤを上記共有鍵で暗号化した第二の暗号化データを上記サーバから受信する受信手段と、

上記第二のワンタイムＩＤを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムＩＤとの照合により、上記サーバを識別し、上記サーバを識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる上記サーバＩＤおよび上記クライアントＩＤに基づいて、上記サーバの正当性を判定する判定手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とする認証システム。

５０．　上記サーバおよび上記クライアントは、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後で、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようになっていることを特徴とする請求項４９に記載の認証システム。

図 1

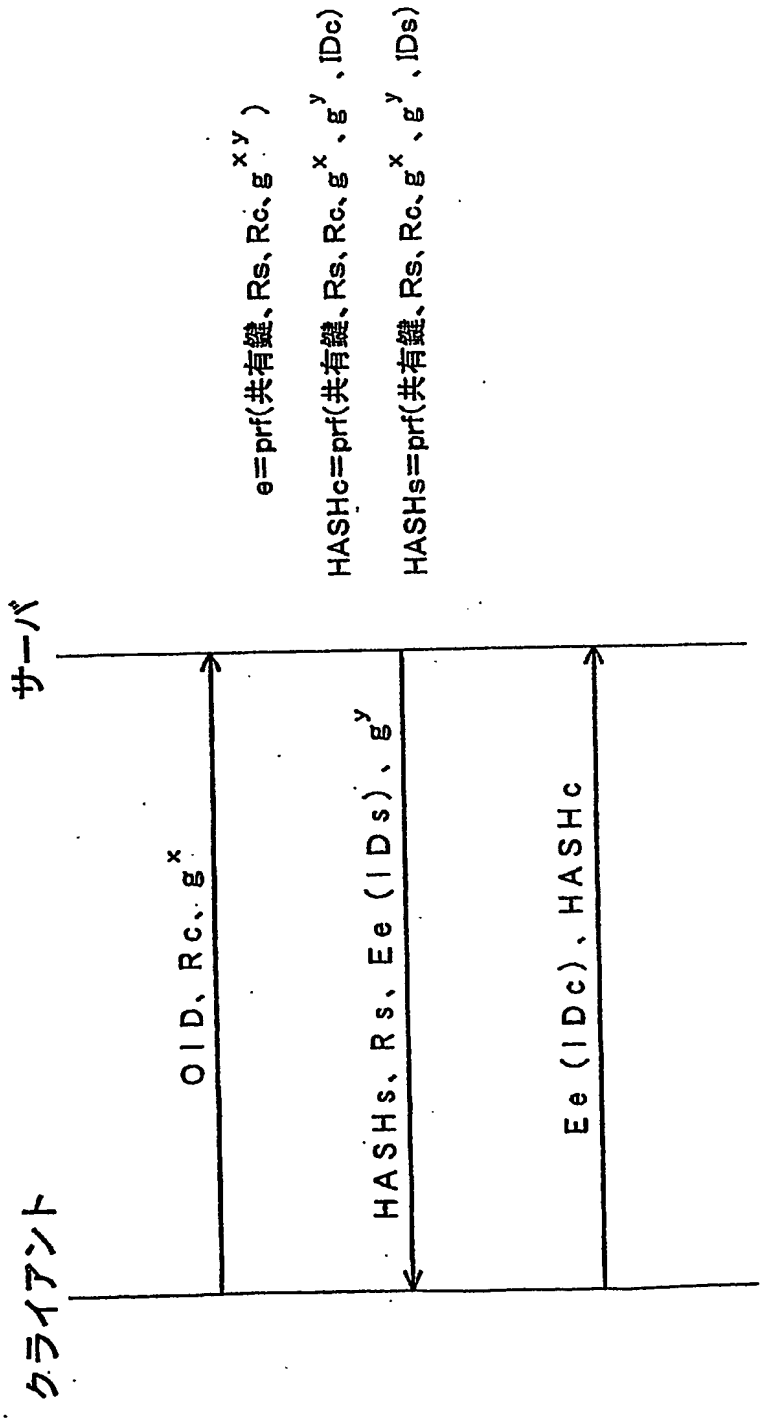

e=prf(共有鍵、Rs、Rc、g^xy)
HASHc=prf(共有鍵、Rs、Rc、g^x、g^y、IDc)
HASHs=prf(共有鍵、Rs、Rc、g^x、g^y、IDs)
サーバ
クライアント
OID、Rc、g^x
HASHs、Rs、Ee(IDs)、g^y
Ee(IDc)、HASHc

図 2

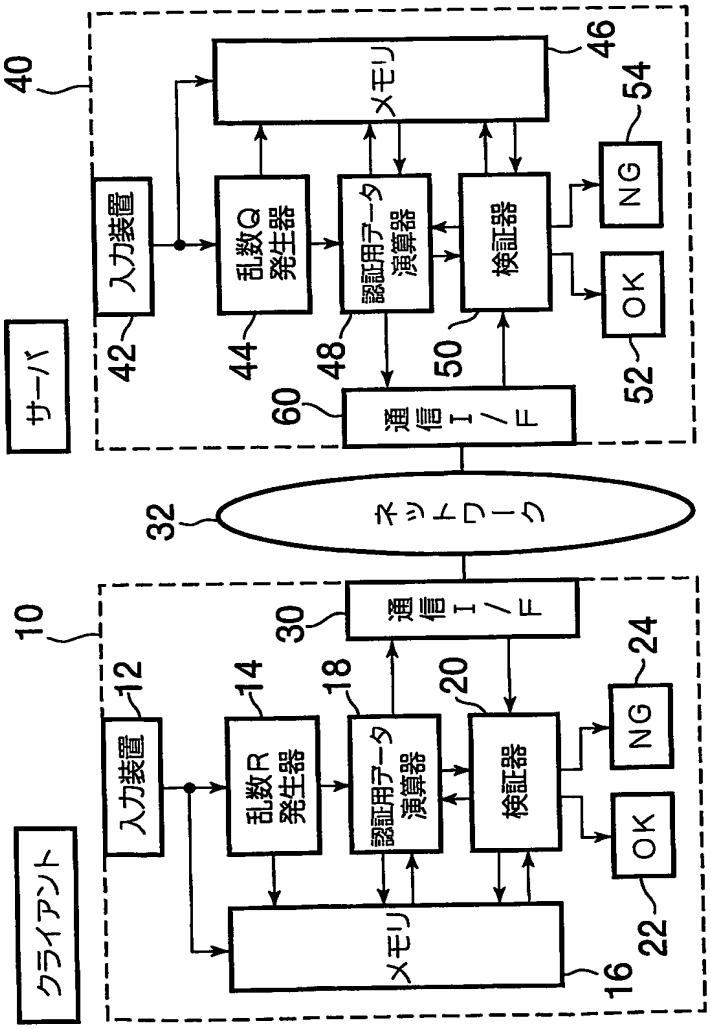

サーバ
40
入力装置
42
乱数Q発生器
44
認証用データ演算器
48
検証器
50
メモリ
46
NG
54
OK
52
通信I/F
60
ネットワーク
32
クライアント
10
入力装置
12
乱数R発生器
14
認証用データ演算器
18
検証器
20
メモリ
16
通信I/F
30
NG
24
OK
22

図 3

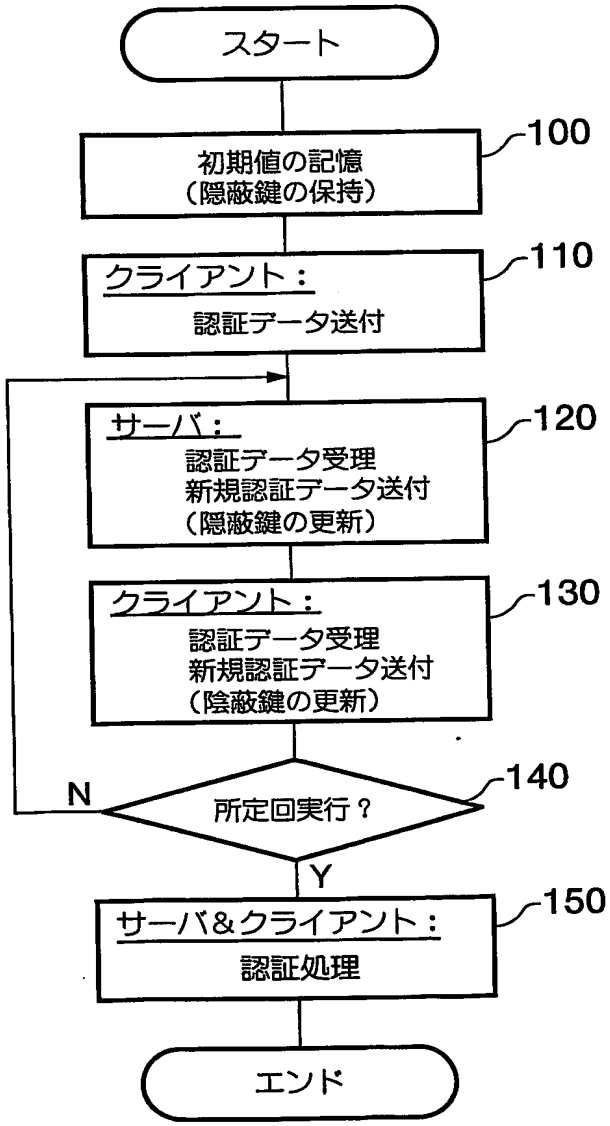

スタート
初期値の記憶
(隠蔽鍵の保持)
100
クライアント：
認証データ送付
110
サーバ：
認証データ受理
新規認証データ送付
(隠蔽鍵の更新)
120
クライアント：
認証データ受理
新規認証データ送付
(陰蔽鍵の更新)
130
N
所定回実行？
140
Y
サーバ&クライアント：
認証処理
150
エンド

図 4

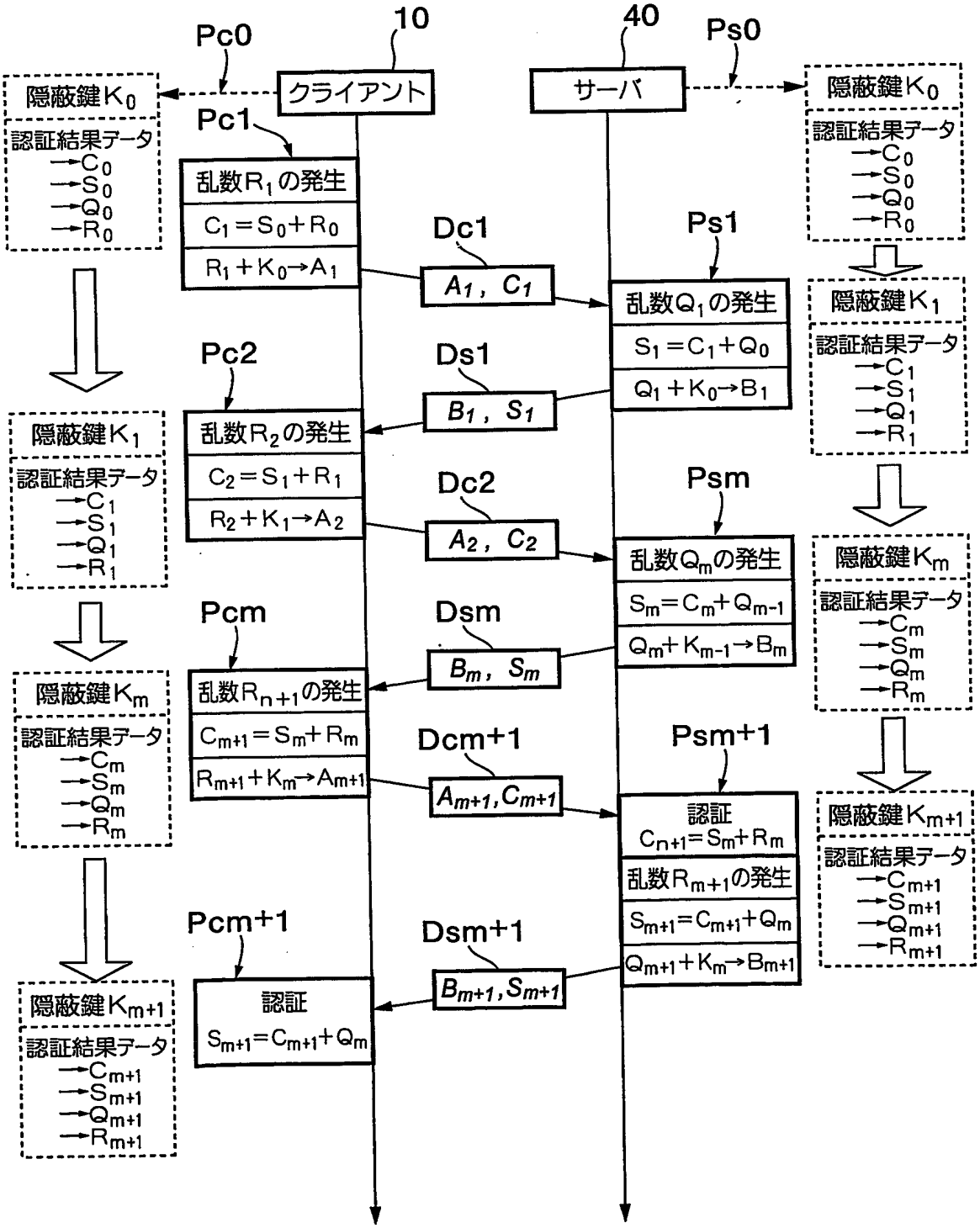

10
40
Pc0
Ps0
クライアント
サーバ
隠蔽鍵K0
認証結果データ
→C0
→S0
→Q0
→R0
Pc1
乱数R1の発生
C1＝S0＋R0
R1＋K0→A1
Dc1
A1, C1
Ps1
乱数Q1の発生
S1＝C1＋Q0
Q1＋K0→B1
隠蔽鍵K1
認証結果データ
→C1
→S1
→Q1
→R1
Pc2
乱数R2の発生
C2＝S1＋R1
R2＋K1→A2
Ds1
B1, S1
Dc2
A2, C2
Psm
乱数Qmの発生
Sm＝Cm＋Qm-1
Qm＋Km-1→Bm
隠蔽鍵Km
認証結果データ
→Cm
→Sm
→Qm
→Rm
Pcm
乱数Rn+1の発生
Cm+1＝Sm＋Rm
Rm+1＋Km→Am+1
Dsm
Bm, Sm
Dcm+1
Am+1, Cm+1
Psm+1
認証
Cn+1＝Sm＋Rm
乱数Rm+1の発生
Sm+1＝Cm+1＋Qm
Qm+1＋Km→Bm+1
隠蔽鍵Km+1
認証結果データ
→Cm+1
→Sm+1
→Qm+1
→Rm+1
Pcm+1
認証
Sm+1＝Cm+1＋Qm
Dsm+1
Bm+1, Sm+1

図 5

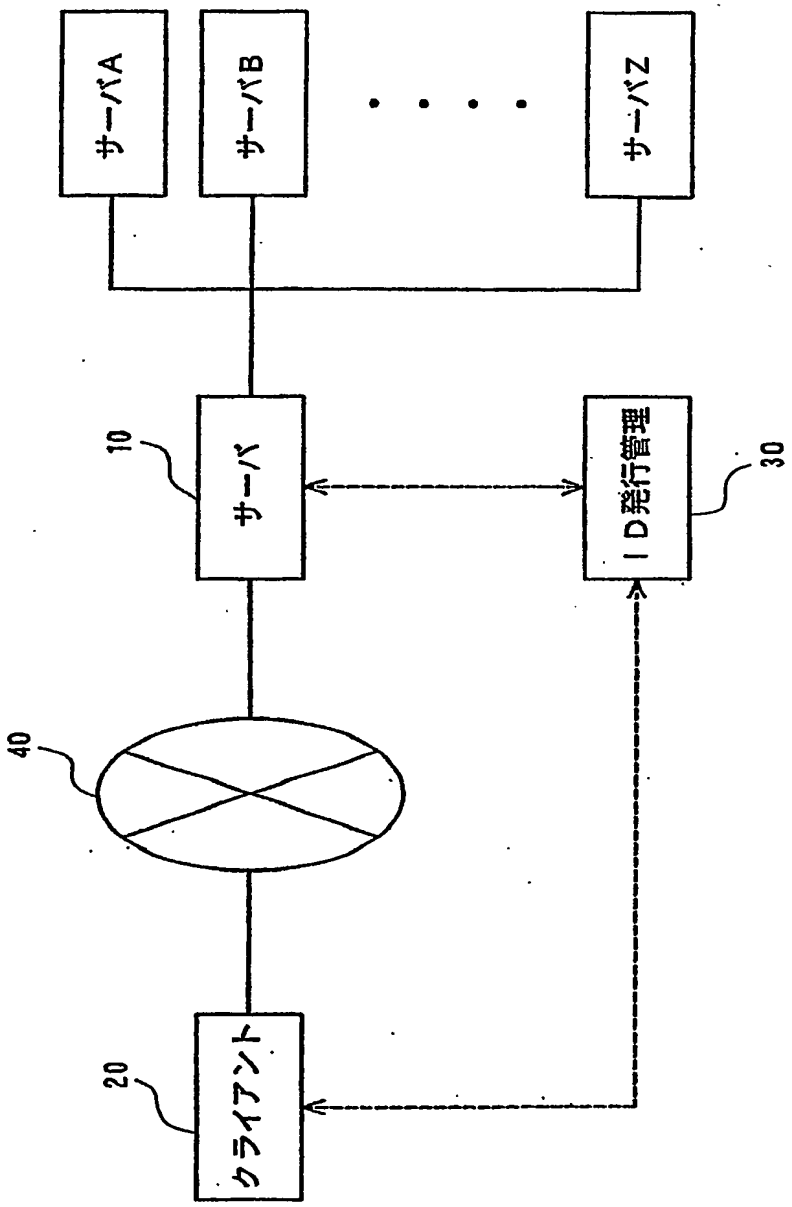


図 6

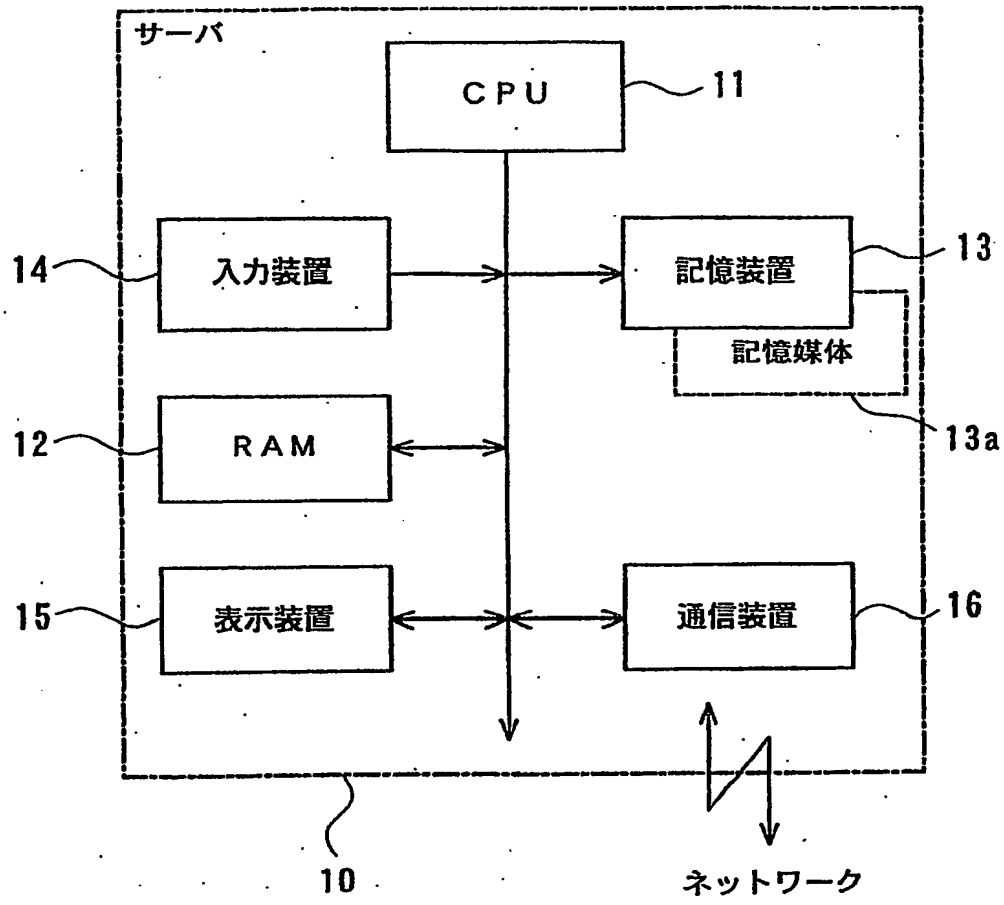

サーバ
CPU
11
14
入力装置
記憶装置
13
記憶媒体
13a
12
RAM
15
表示装置
通信装置
16
10
ネットワーク

図 7

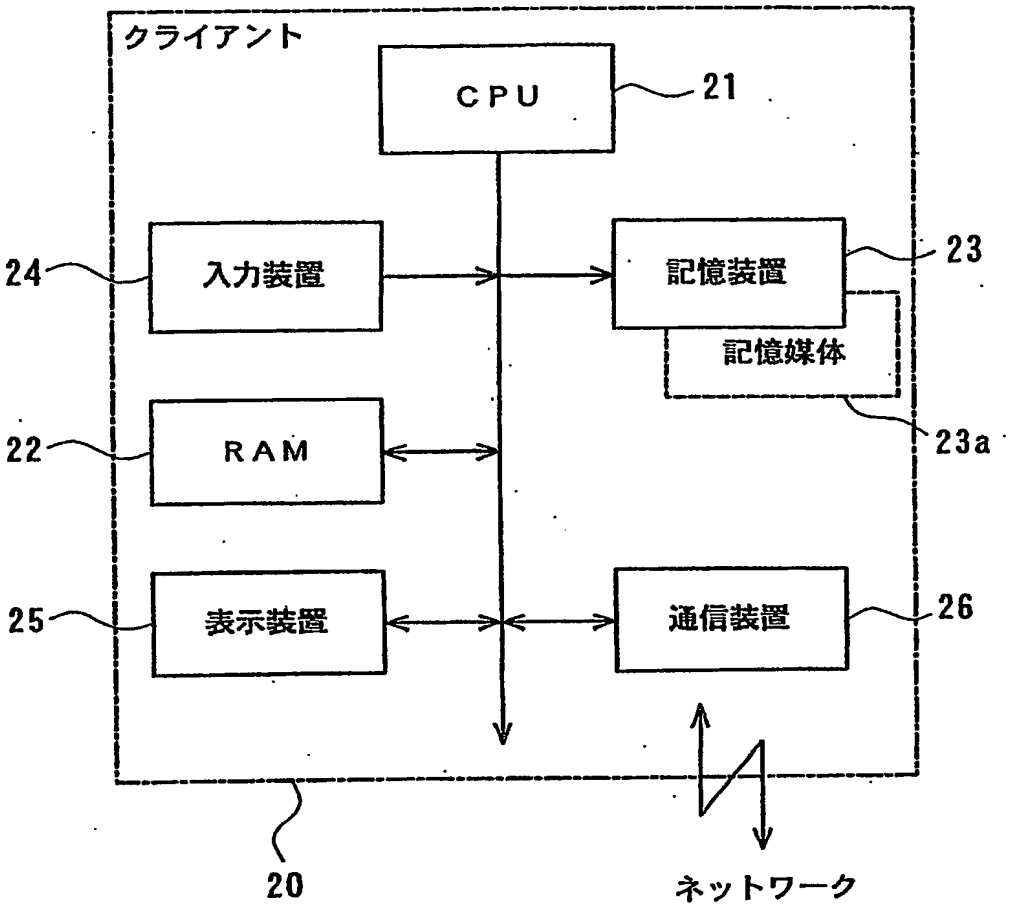

クライアント
CPU
21
24
入力装置
記憶装置
23
記憶媒体
23a
22
RAM
25
表示装置
通信装置
26
20
ネットワーク

図 8

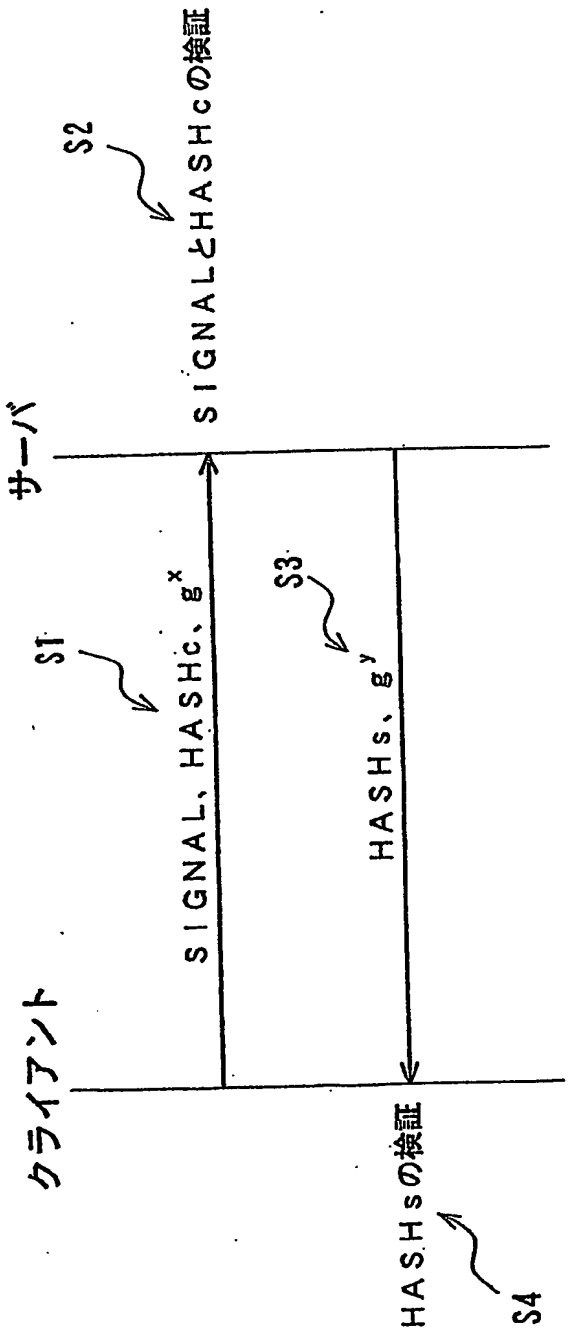

S2
SIGNALとHASHcの検証
サーバ
S1
SIGNAL、HASHc、g^x
S3
HASHs、g^y
クライアント
HASHsの検証
S4

図 9

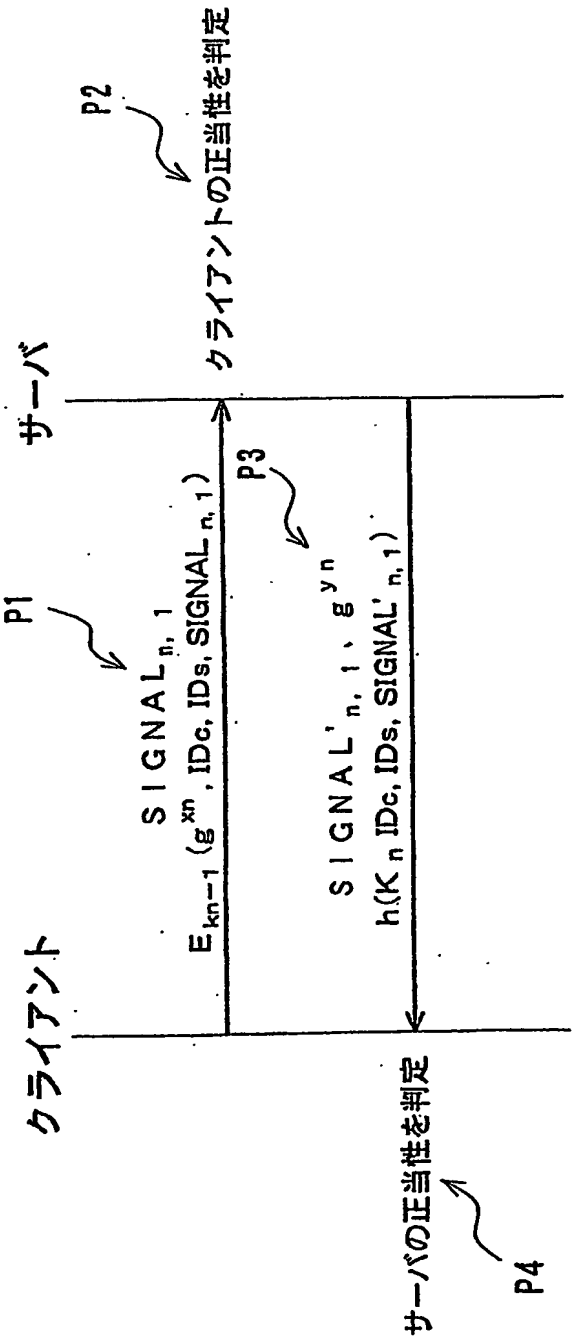

P2
クライアントの正当性を判定
サーバ
P1
SIGNAL n, 1
E kn−1 (g^xn, IDc, IDs, SIGNAL n, 1)
P3
SIGNAL' n, 1、g^yn
h(K n IDc, IDs, SIGNAL' n, 1)
クライアント
サーバの正当性を判定
P4

図 10

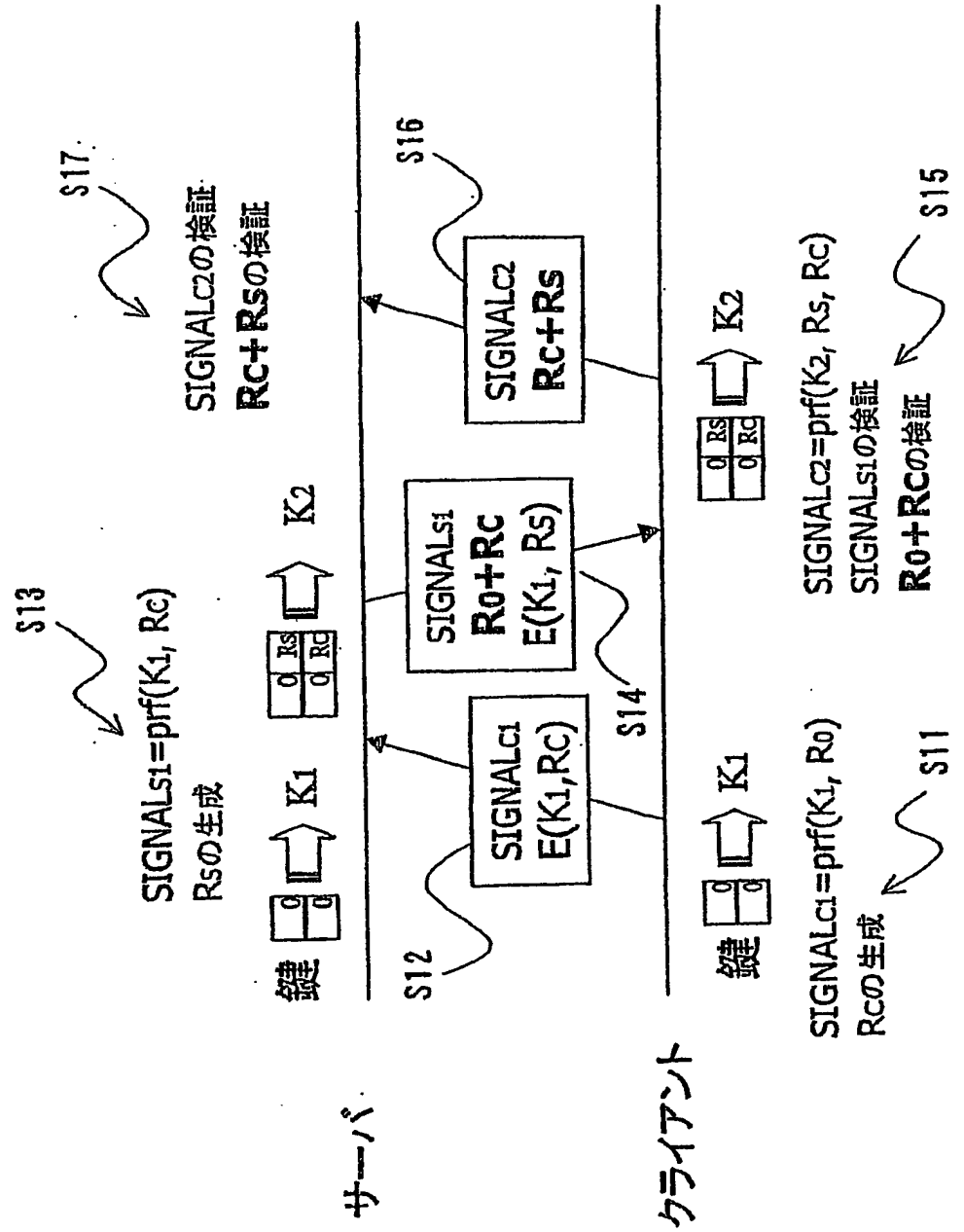

S17
SIGNALc2の検証
Rc+Rsの検証
S16
SIGNALc2
Rc+Rs
S13
SIGNALs1=prf(K1, Rc)
Rsの生成
K2
Rs
Rc
K1
鍵
SIGNALs1
R0+Rc
E(K1, Rs)
S12
SIGNALc1
E(K1,Rc)
S14
サーバ
クライアント
K2
Rs
Rc
K1
鍵
SIGNALc2=prf(K2, Rs, Rc)
SIGNALs1の検証
R0+Rcの検証
S15
SIGNALc1=prf(K1, R0)
Rcの生成
S11

図 11

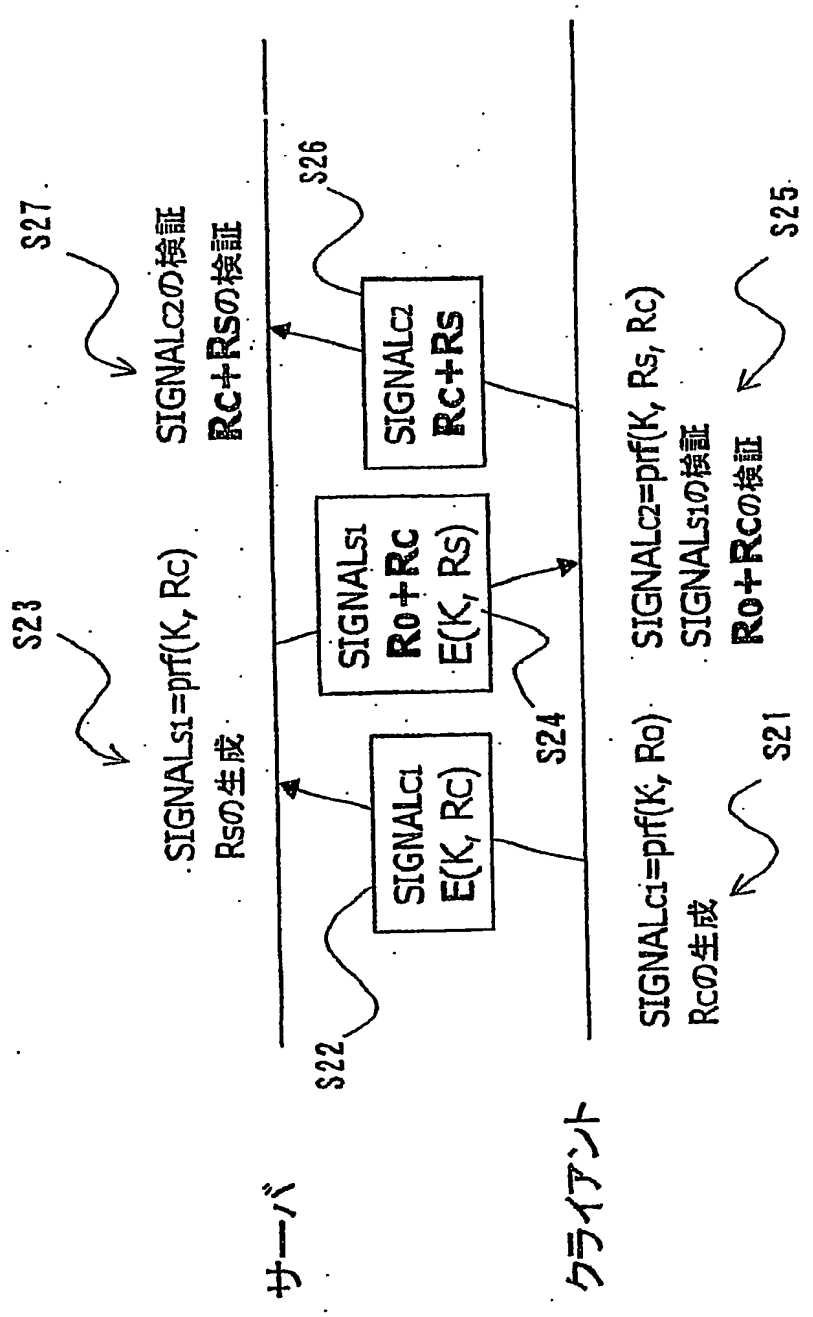

サーバ
クライアント
S21
SIGNALc1=prf(K, Ro)
Rcの生成
S22
SIGNALc1
E(K, Rc)
S23
SIGNALs1=prf(K, Rc)
Rsの生成
S24
SIGNALs1
Ro+Rc
E(K, Rs)
S25
SIGNALc2=prf(K, Rs, Rc)
SIGNALs1の検証
Ro+Rcの検証
S26
SIGNALc2
Rc+Rs
S27
SIGNALc2の検証
Rc+Rsの検証

図 12

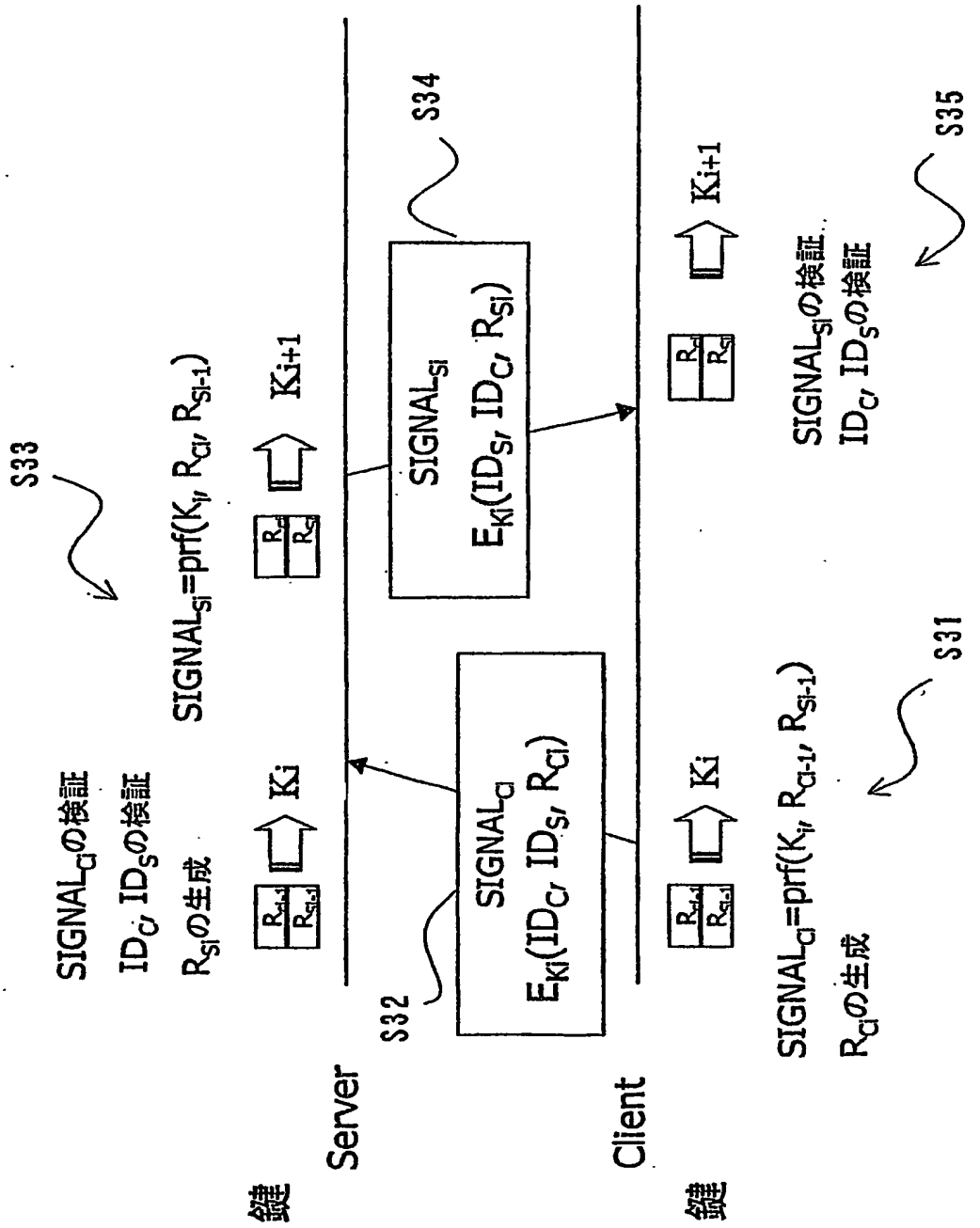

S31
S32
S33
S34
S35
Server
Client
鍵
鍵
SIGNAL_Ci=prf(K_i, R_Ci-1, R_Si-1)
R_Ciの生成
SIGNAL_Ci
E_Ki(ID_C, ID_S, R_Ci)
SIGNAL_Siの検証
ID_C, ID_Sの検証
SIGNAL_Si
E_Ki(ID_S, ID_C, R_Si)
SIGNAL_Ciの検証
ID_C, ID_Sの検証
R_Siの生成
SIGNAL_Si=prf(K_i, R_Ci, R_Si-1)
Ki
Ki+1

図 13

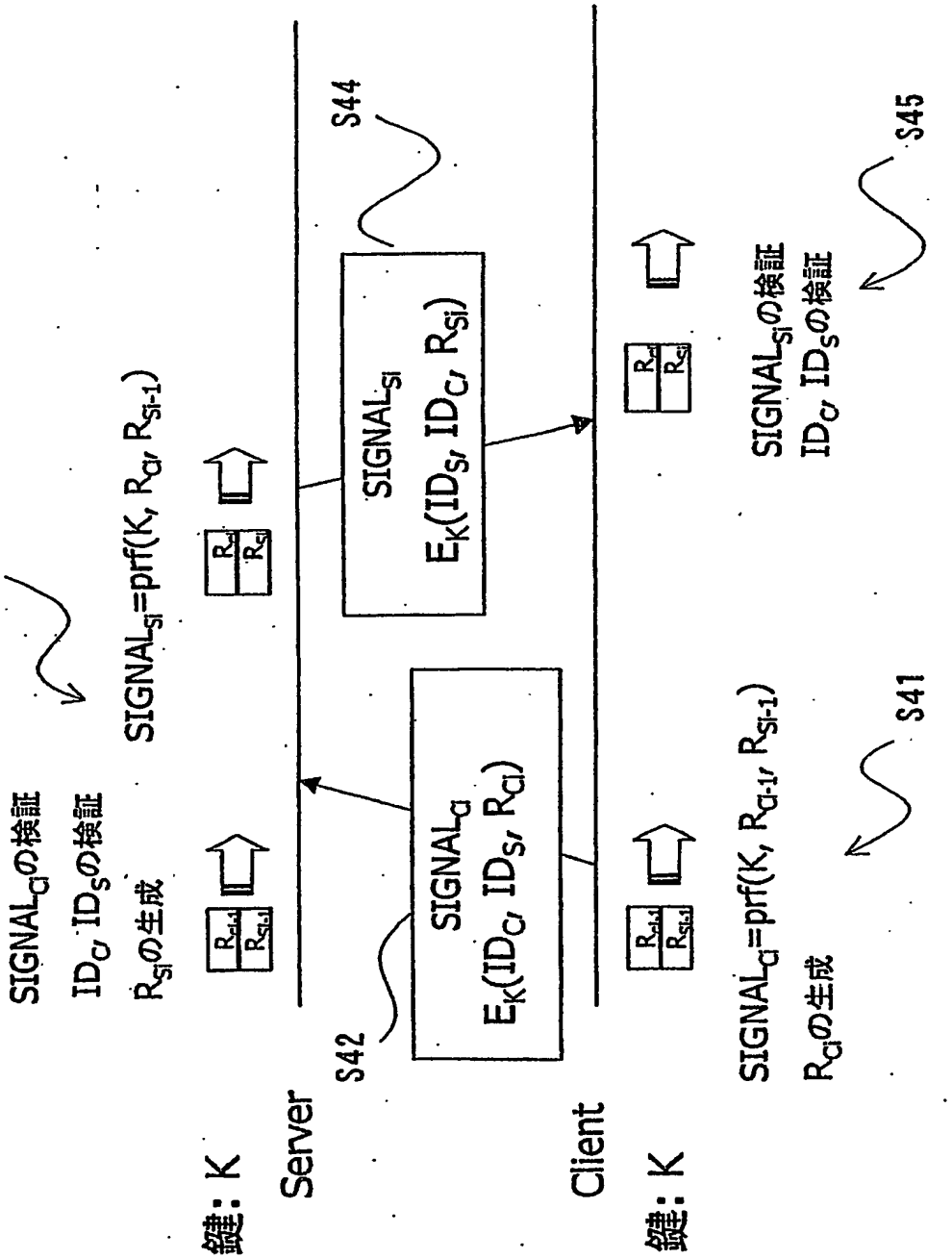

S41
SIGNAL_Ci=prf(K, R_Ci-1, R_Si-1)
R_Ci の生成
鍵：K
Client
S42
SIGNAL_Ci
E_K(ID_C, ID_S, R_Ci)
S43
SIGNAL_Ci の検証
ID_C, ID_S の検証
R_Si の生成
SIGNAL_Si=prf(K, R_Ci, R_Si-1)
鍵：K
Server
S44
SIGNAL_Si
E_K(ID_S, ID_C, R_Si)
S45
SIGNAL_Si の検証
ID_C, ID_S の検証

図 14

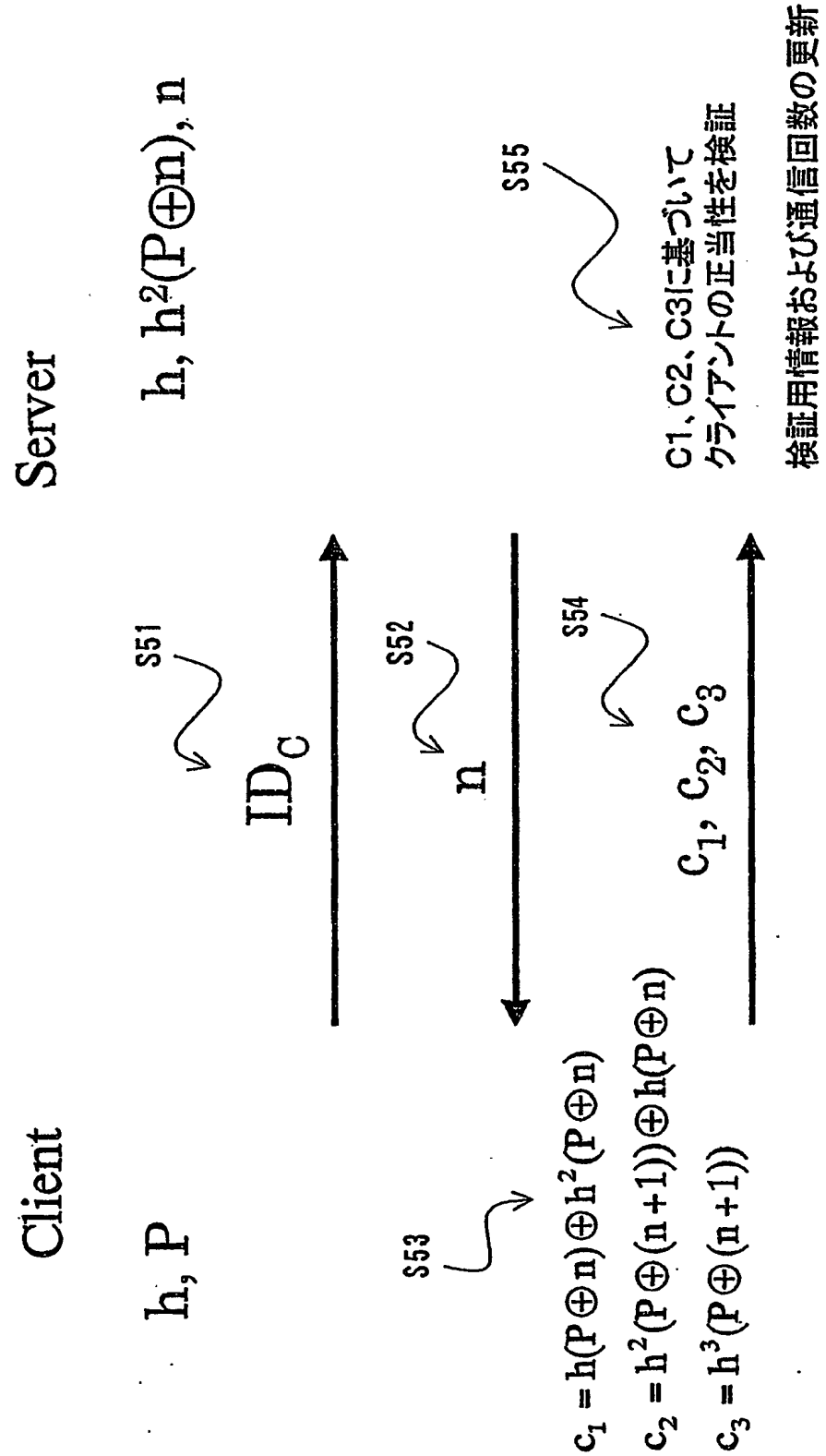

Client
Server
h, P
h, h²(P⊕n), n
S51
ID_C
S52
n
S53
S54
S55
c1 = h(P⊕n)⊕h²(P⊕n)
c2 = h²(P⊕(n+1))⊕h(P⊕n)
c3 = h³(P⊕(n+1))
C1, C2, C3
C1、C2、C3に基づいて
クライアントの正当性を検証
検証用情報および通信回数の更新

図 15

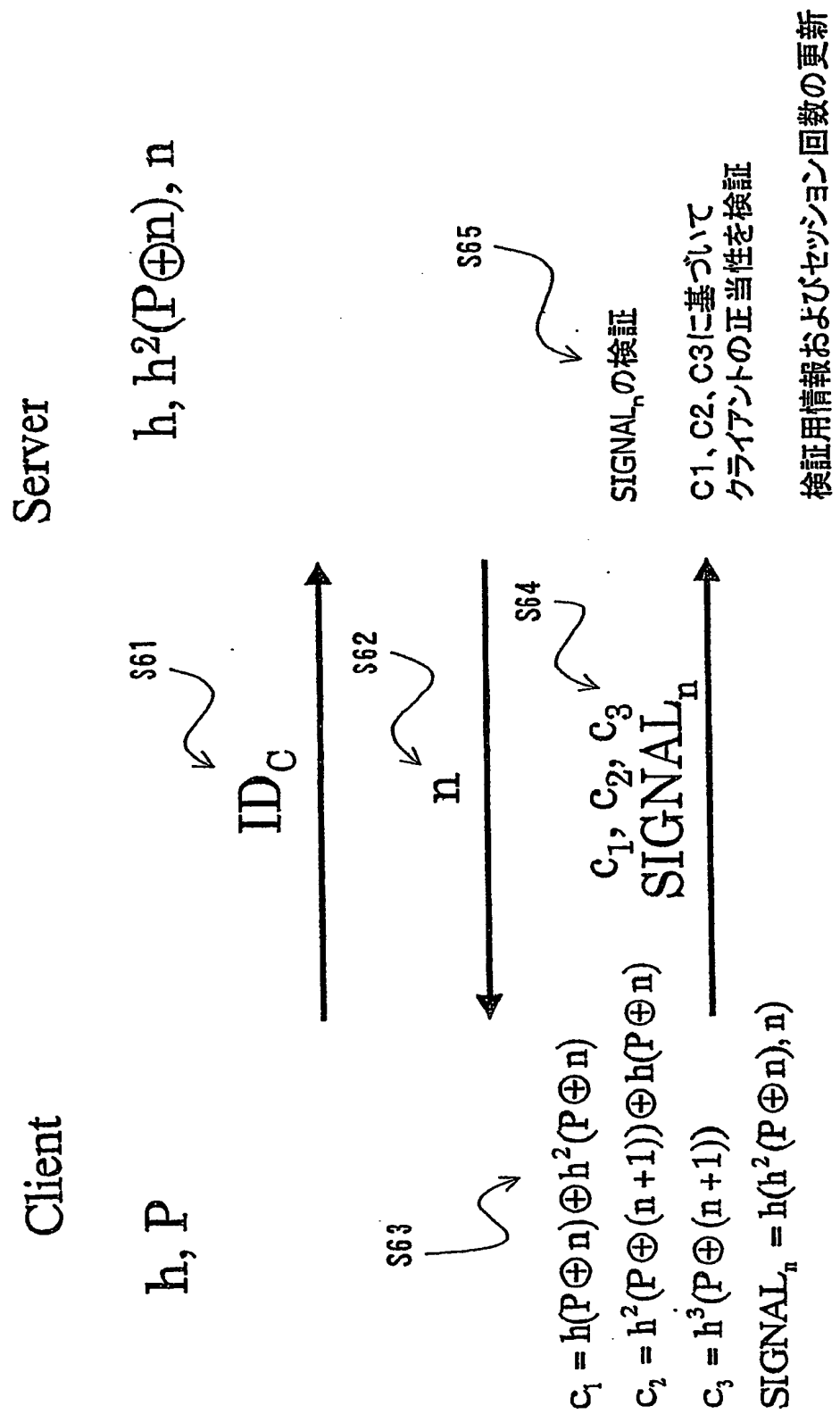
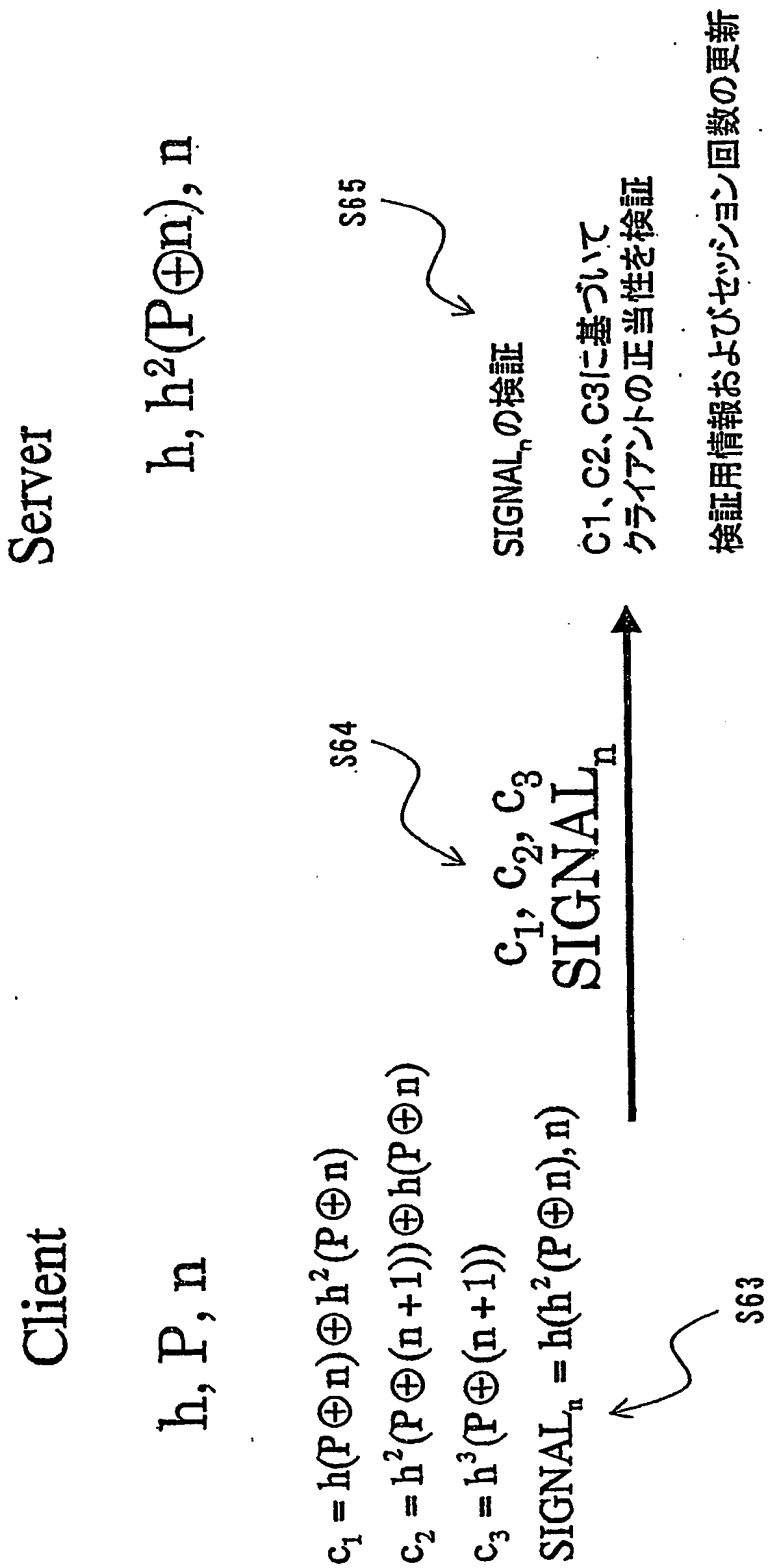

Client
Server
h, P
h, h²(P⊕n), n
S61
ID_C
S62
n
S63
S64
c1, c2, c3
SIGNAL_n
S65
SIGNAL_nの検証
C1、C2、C3に基づいて
クライアントの正当性を検証
検証用情報およびセッション回数の更新
c1 = h(P⊕n)⊕h²(P⊕n)
c2 = h²(P⊕(n+1))⊕h(P⊕n)
c3 = h³(P⊕(n+1))
SIGNAL_n = h(h²(P⊕n), n)

図 16


Client
h, P, n
$c_1 = h(P \oplus n) \oplus h^2(P \oplus n)$
$c_2 = h^2(P \oplus (n+1)) \oplus h(P \oplus n)$
$c_3 = h^3(P \oplus (n+1))$
$SIGNAL_n = h(h^2(P \oplus n), n)$
S63
S64
c_1, c_2, c_3
$SIGNAL_n$
Server
h, $h^2(P \oplus n)$, n
S65
$SIGNAL_n$の検証
C1、C2、C3に基づいて
クライアントの正当性を検証
検証用情報およびセッション回数の更新

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/07794

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[7] H04L9/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[7] H04L9/32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2003
Kokai Jitsuyo Shinan Koho 1971-2003 Jitsuyo Shinan Toroku Koho 1996-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
JICST FILE(JOIS), WPI, one-time, identity

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	HANDBOOK of APPLIED CRYPTOGRAPHY, CRC Press, 1997, pages 400 to 403, especially (ii) Challenge-response based on(keyed) one-way functions	16,19,25, 44-46
A	Smafat, D. & Molva, R., A method providing identity privacy to mobile users during authentication, Proceedings of Workshop on Mobile Computing Systems and Applications, 1995, pages 196 to 199, especially 4,2 Alias Computation	14-26,28-50

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
21 November, 2003 (21.11.03)

Date of mailing of the international search report
09 December, 2003 (09.12.03)

Name and mailing address of the ISA/
Japanese Patent Office

Facsimile No.

Authorized officer

Telephone No.

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP03/07794

Box I Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☒ Claims Nos.: 1-13, 27
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
See extra sheet for claims 1-13.
As for claim 27 (authentication method), it is not described to which claim it refers.

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

The inventions of claims 1-13 relate to inter-authentication not related to the one time ID. The inventions of claims 14-26, 28-50 relate to the one time ID. It should be noted that since the inter-authentication is a known technique and cannot be a "special technical feature".

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☒ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

Continuation of Box No.I-2 of continuation of first sheet(1)

In claim 1, it is unknown whether the "new storage data generated" in the description "the new storage data generated is encrypted by using the history data and transmitted to the second authentication device" is the same data as the "storage data" in the description "the first authentication device generates new storage data by using the history data stored". Similarly, it is unknown whether the "new storage data generated" in the description "the new storage data generated is encrypted by using the history data and transmitted to the first authentication device" is the same data as the "storage data" in the description "the second authentication device generates new storage data by using the history data stored". Moreover, if they are different data, it is unknown what kind of meaning is present in generating storage data by using the history data. Similarly, it is unknown whether the "storage data from the first authentication device" in the description "the second authentication device generates new storage data by using storage data from the first authentication device" is the same data as the "storage data" in the description "first transmission step for encrypting the new storage data generated and transmitting it to the second authentication device". If they are different data, it is unknown what kind of data is the "storage data from the first authentication device".

Furthermore, there are descriptions that "the first authentication device generates new storage data by using the history data stored" and "the second authentication device generates new storage data by using the history data stored". In each of these descriptions, it is unknown whether the data is the same data as the "history data" in the description "using as the history data the updated result which has been updated by using the storage data by the previous authentication." If they are different data, it is unknown what kind of data is "history data" in these descriptions and "said history data" in the subsequent description.

Moreover, it is unknown what kind of data is the "storage data by the previous authentication" in the description "using as the history data the updated result which has been updated by using the storage data by the previous authentication".

That is, it is unknown how the "inter-authentication method" is performed because the relationship between the plurality of "storage data" and "history data" is unclear. The same applies to claims 2-13 which refer to claim 1.

国際調査報告

国際出願番号 PCT／JP03／07794

A. 発明の属する分野の分類（国際特許分類（IPC））

Int. Cl⁷ H04L9／32

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（IPC））

Int. Cl⁷ H04L9／32

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922－1996年
日本国公開実用新案公報	1971－2003年
日本国登録実用新案公報	1994－2003年
日本国実用新案登録公報	1996－2003年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

JICSTファイル（JOIS），WPI
one-time, identity

C. 関連すると認められる文献

引用文献のカテゴリー＊	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
X	HANDBOOK of APPLIED CRYPTOGRAPHY, CRC Press, 1997, p. 400-403 especially (ii) Challenge-response based on(keyed)one-way functions	16, 19, 25, 44-46
A	Samfat, D. & Molva, R., A method providing identity privacy to mobile users during authentication, Proceedings of Workshop on Mobile Computing Systems and Applications, 1995, p. 196-199, especially 4,2 Alias Computation	14-26, 28-50

☐ C欄の続きにも文献が列挙されている。

☐ パテントファミリーに関する別紙を参照。

＊ 引用文献のカテゴリー
「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日 21. 11. 03	国際調査報告の発送日 09.12.03
国際調査機関の名称及びあて先 日本国特許庁（ISA／JP） 郵便番号100－8915 東京都千代田区霞が関三丁目4番3号	特許庁審査官（権限のある職員） 中里 裕正 5M 9364 電話番号 03－3581－1101 内線 3597

様式PCT／ISA／210（第2ページ）（1998年7月）

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ０３／０７７９４

第Ⅰ欄　請求の範囲の一部の調査ができないときの意見（第１ページの２の続き）

法第８条第３項（ＰＣＴ１７条(2)(a)）の規定により、この国際調査報告は次の理由により請求の範囲の一部について作成しなかった。

1．☐ 請求の範囲 _____ は、この国際調査機関が調査をすることを要しない対象に係るものである。
つまり、

2．☒ 請求の範囲 １－１３，２７ は、有意義な国際調査をすることができる程度まで所定の要件を満たしていない国際出願の部分に係るものである。つまり、
請求の範囲１－１３については、別紙を参照されたい。
請求の範囲２７には、「請求項に記載の認証方法」と記載されており、いずれの請求項を引用しているのか不明である。

3．☐ 請求の範囲 _____ は、従属請求の範囲であってＰＣＴ規則6.4(a)の第２文及び第３文の規定に従って記載されていない。

第Ⅱ欄　発明の単一性が欠如しているときの意見（第１ページの３の続き）

次に述べるようにこの国際出願に二以上の発明があるとこの国際調査機関は認めた。

請求の範囲１－１３は、ワンタイムＩＤとは何ら関係しない相互認証に関する発明であり、請求の範囲１４－２６，２８－５０は、ワンタイムＩＤに関する発明である。なお、相互認証は周知の技術であるから、相互認証を「特別な技術的特徴」とすることはできない。

1．☐ 出願人が必要な追加調査手数料をすべて期間内に納付したので、この国際調査報告は、すべての調査可能な請求の範囲について作成した。

2．☒ 追加調査手数料を要求するまでもなく、すべての調査可能な請求の範囲について調査することができたので、追加調査手数料の納付を求めなかった。

3．☐ 出願人が必要な追加調査手数料を一部のみしか期間内に納付しなかったので、この国際調査報告は、手数料の納付のあった次の請求の範囲のみについて作成した。

4．☐ 出願人が必要な追加調査手数料を期間内に納付しなかったので、この国際調査報告は、請求の範囲の最初に記載されている発明に係る次の請求の範囲について作成した。

追加調査手数料の異議の申立てに関する注意
☐ 追加調査手数料の納付と共に出願人から異議申立てがあった。
☐ 追加調査手数料の納付と共に出願人から異議申立てがなかった。

請求の範囲１の「生成した新規の記憶データを前記履歴データを用いて暗号化して第２認証装置に送信する」との記載における「生成した新規の記憶データ」が、「前記第１認証装置は、記憶されている履歴データを用いて記憶データを新規に生成し」との記載における「記憶データ」と同一のデータであるのか否か不明であり、同様に、「生成した新規の記憶データを前記履歴データを用いて暗号化して第１認証装置に送信する」との記載における「生成した新規の記憶データ」が、「前記第２認証装置は、…記憶されている履歴データを用いて記憶データを新規に生成し」との記載における「記憶データ」と同一のデータであるのか否かという点も不明である。また、それぞれ別個のデータであるならば、履歴データを用いて記憶データを生成することにどのような意味があるのか不明である。同じく、「前記第２認証装置は、前記第１認証装置からの記憶データ…を用いて新規に記憶データを生成し」との記載における「第１認証装置からの記憶データ」とは、「生成した新規の記憶データを…暗号化して第２の認証装置に送信する第１送信工程」との記載における「記憶データ」と同一のデータであるのか否か不明であり、異なるデータであるならば、「第１認証装置からの記憶データ」とは、如何なるデータであるのか不明である。

さらに「前記第１認証装置は、記憶されている履歴データを用いて記憶データを新規に生成し」及び「前記第２認証装置は、…記憶されている履歴データを用いて記憶データを新規に生成し」と記載されているが、各々、「前回の認証による記憶データを用いて更新した更新結果を履歴データとして」との記載における「履歴データ」と同一のデータであるのか不明であり、異なるデータであるならば、これらの記載における「履歴データ」及びその後の記載における「前記履歴データ」とは、如何なるデータであるのか不明である。

また「前回の認証による記憶データを用いて更新した更新結果を履歴データとして」との記載における「前回の認証による記憶データ」とは如何なるものかということも不明である。

すなわち請求の範囲１に記載された「相互認証方法」については、複数存在する「記憶データ」及び「履歴データ」の間の関係が不明であるため、どのような手順にて認証が行われているのか把握することができない。請求の範囲１を引用する請求の範囲２～１３についても同様である。